



## AlphaServer GS80/160/320

### User's Guide

Order Number: EK-GS320-UG. B01

This guide is intended for those who manage, operate, or service the *AlphaServer* GS160/320 system and the *AlphaServer* GS80 rack system. It covers configuration guidelines, operation, system management, and basic troubleshooting.

---

**First Printing, May 2000**

© 2000 Compaq Computer Corporation.

COMPAQ and the Compaq logo registered in U.S. Patent and Trademark Office. AlphaServer, OpenVMS, StorageWorks, and Tru64 are trademarks of Compaq Information Technologies Group, L.P.

Portions of the software are © copyright Cimetrix Technology. Linux is a registered trademark of Linus Torvalds in several countries. UNIX is a registered trademark of The Open Group in the U.S. and other countries. All other product names mentioned herein may be trademarks of their respective companies.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

**FCC Notice:** The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference. Any modifications to this device—unless expressly approved by the manufacturer—can void the user's authority to operate this equipment under part 15 of the FCC rules.

**Shielded Cables:** If shielded cables have been supplied or specified, they must be used on the system in order to maintain international regulatory compliance.

**Warning!** This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

**Achtung!** Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

**Attention!** Ceci est un produit de Classe A. Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre mesures spécifiques appropriées.

**警告使用者:**

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

# Contents

<b>Preface .....</b>	<b>ix</b>
----------------------	-----------

## **Chapter 1 Introduction**

1.1	AlphaServer GS160/320 and GS80 Systems.....	1-2
1.2	Firmware Utilities Overview .....	1-4
1.3	System Architecture.....	1-5

## **Chapter 2 GS160/320 System Overview**

2.1	System Characteristics .....	2-2
2.2	System Box Architecture .....	2-4
2.3	Quad Building Block (QBB) Components .....	2-6
2.3.1	Backplane .....	2-8
2.3.2	CPU Module.....	2-10
2.3.3	Memory Module .....	2-11
2.3.4	Directory Module .....	2-12
2.3.5	Power Modules.....	2-13
2.3.6	Power System Manager Module .....	2-14
2.3.7	Clock Splitter Module .....	2-15
2.3.8	I/O Riser Module.....	2-16
2.3.9	Global Port Module .....	2-17
2.3.10	Distribution Board.....	2-18
2.3.11	Hierarchical Switch .....	2-20
2.4	Power System.....	2-22
2.4.1	AC Input Box .....	2-24
2.5	PCI I/O .....	2-26
2.6	Control Panel .....	2-28
2.6.1	Control Panel LEDs.....	2-30

## **Chapter 3 GS160/320 System Configuration Rules**

3.1	GS160 System Cabinet .....	3-2
3.2	GS320 System Cabinets.....	3-4
3.3	Power Cabinet.....	3-6
3.3.1	Power Supply Slot Assignments .....	3-8
3.4	System Box.....	3-10
3.5	QBB Color Code .....	3-12
3.6	Memory Configurations .....	3-14
3.7	Memory Interleaving Guidelines .....	3-16
3.8	PCI Boxes.....	3-18
3.9	PCI Box Slot Configuration.....	3-20
3.10	Expander Cabinet .....	3-22

## **Chapter 4 GS80 Rack System Overview**

4.1	Rack System Characteristics .....	4-2
4.2	System Drawer Architecture .....	4-4
4.3	System Drawer Modules .....	4-6

## **Chapter 5 GS80 Rack System Configuration Rules**

5.1	Rack.....	5-2
5.2	Rack Power System.....	5-4

## **Chapter 6 Booting and Installing an Operating System**

6.1	Powering Up the System.....	6-2
6.1.1	SCM Power-Up Display .....	6-4
6.1.2	SRM Power-Up Display .....	6-6
6.2	Setting Boot Options .....	6-10
6.2.1	Bootdef_dev.....	6-10
6.2.2	Boot_file .....	6-11
6.2.3	Boot_osflags .....	6-12
6.2.4	ei*0_inet_init or ew*0_inet_init.....	6-14
6.2.5	ei*0_protocols or ew*0_protocols .....	6-15
6.3	Booting Tru64 UNIX.....	6-16
6.3.1	Booting Tru64 UNIX Over the Network.....	6-18
6.4	Installing Tru64 UNIX.....	6-20
6.5	Booting OpenVMS.....	6-22
6.5.1	Booting OpenVMS from the InfoServer.....	6-24
6.6	Installing OpenVMS .....	6-26

## Chapter 7 Operation

7.1	SRM Console .....	7-2
7.1.1	SRM Command Overview .....	7-4
7.1.2	Setting the Control Panel Message .....	7-6
7.2	Displaying the System Configuration .....	7-7
7.2.1	Show Boot Command .....	7-7
7.2.2	Show Config Command .....	7-8
7.2.3	Show Device Command .....	7-16
7.2.4	Show Memory Command .....	7-18
7.3	Setting SRM Environment Variables .....	7-19
7.4	Setting SRM Console Security .....	7-20
7.4.1	Setting Tru64 UNIX or OpenVMS Systems to Auto Start .....	7-21
7.5	Changing the Default Boot Device .....	7-21
7.6	Soft Partitioning .....	7-22
7.7	Hard Partitioning .....	7-26

## Chapter 8 Using the System Control Manager

8.1	Console Serial Bus Subsystem .....	8-2
8.2	System Control Manager Overview .....	8-4
8.3	SCM COM1 Operating Modes .....	8-6
8.3.1	Bypass Modes .....	8-8
8.4	Console Device Setup .....	8-10
8.5	Entering the SCM .....	8-12
8.6	SRM Environment Variables for COM1 .....	8-13
8.7	SCM Command-Line Interface .....	8-14
8.7.1	Defining the COM1 Data Flow .....	8-17
8.7.2	Displaying the System Status .....	8-18
8.7.3	Displaying the System Environment .....	8-20
8.7.4	Power On and Off, Reset, and Halt .....	8-22
8.7.5	Configuring Remote Dial-In .....	8-24
8.7.6	Configuring Alert Dial-Out .....	8-26
8.7.7	Resetting the Escape Sequence .....	8-29
8.8	Troubleshooting Tips .....	8-30

## Appendix A Jumpering Information

A.1	PCI Backplane Jumpers .....	A-1
A.2	HPM Jumpers .....	A-2
A.3	Standard I/O Module Jumpers .....	A-2

## Glossary

## Index

## Examples

6-1	SCM Power-Up Display .....	6-2
6-2	SRM Power-Up Display .....	6-6
6-3	Booting Tru64 UNIX from a Local SCSI Disk .....	6-16
6-4	RIS Boot .....	6-18
6-5	Tru64 UNIX Installation Display .....	6-20
6-6	Booting OpenVMS from a Local Disk .....	6-22
6-7	InfoServer Boot .....	6-24
6-8	OpenVMS Installation Menu .....	6-26
7-1	Set Ocp_Text Command.....	7-6
7-2	Show Boot* .....	7-7
7-3	Show Config .....	7-8
7-4	Show Device .....	7-16
7-5	Show Memory.....	7-18
7-6	Defining Soft Partitions .....	7-24
7-7	Defining Hard Partitions .....	7-26
8-1	Dial-In Configuration.....	8-24
8-2	Alert Dial-Out Configuration.....	8-26

## Figures

1-1	AlphaServer GS160 and GS80 Systems.....	1-2
1-2	Sample System Architecture.....	1-5
2-1	System Box.....	2-2
2-2	System Box Block Diagram (16-Processor System) .....	2-4
2-3	System Box Block Diagram (8-Processor System) .....	2-5
2-4	System Box QBBs (Top View) .....	2-6
2-5	Backplane (System Box, Front View) .....	2-8
2-6	CPU Module .....	2-10
2-7	Memory Module .....	2-11
2-8	Directory Module.....	2-12
2-9	Power Modules .....	2-13
2-10	Power System Manager Module .....	2-14
2-11	Clock Splitter Module .....	2-15
2-12	I/O Riser Module .....	2-16
2-13	Global Port Module .....	2-17

2-14	Distribution Board .....	2-18
2-15	Distribution Board in Single-Box System.....	2-19
2-16	Hierarchical Switch.....	2-20
2-17	Power System.....	2-22
2-18	AC Input Box .....	2-24
2-19	PCI Master Box.....	2-26
2-20	Control Panel .....	2-28
2-21	Control Panel LED Status .....	2-30
3-1	GS160 System .....	3-2
3-2	GS320 System .....	3-4
3-3	Power Cabinet Configuration (32-P System) .....	3-6
3-4	Power Supply Slot Assignments .....	3-8
3-5	System Box QBB (Cabinet Front).....	3-10
3-6	System Box QBB (Cabinet Rear) .....	3-11
3-7	QBB Center Bar Color Code (Cabinet Front) .....	3-12
3-8	QBB Center Bar Color Code (Cabinet Rear).....	3-13
3-9	Memory Module and Directory Module .....	3-14
3-10	Sample I/O Subsystem .....	3-18
3-11	PCI Slot Locations.....	3-20
3-12	BA356 Storage Device Configurations .....	3-22
4-1	System Drawer.....	4-2
4-2	Two-Drawer Block Diagram .....	4-4
4-3	System Drawer Modules .....	4-6
4-4	System Drawer Backplane.....	4-7
5-1	Rack.....	5-2
5-2	Two-Drawer Rack Power System.....	5-4
8-1	CSB Block Diagram .....	8-2
8-2	Data Flow in Through Mode .....	8-6
8-3	Data Flow in Bypass Mode .....	8-8
8-4	Setups for SCM (PCI Box).....	8-11

## Tables

1	AlphaServer 80/160/320 Family Documentation .....	xiv
2-1	System Box Characteristics .....	2-2
2-2	Power Cabinet and System Environmental Characteristics .....	2-3
2-3	AC Input Box Circuit Breakers.....	2-25
3-1	Interleaving Memory Modules .....	3-16
3-2	PCI Slots and Logical Hoses .....	3-21
4-1	System Drawer Characteristics .....	4-2
4-2	Rack System Characteristics .....	4-3
6-1	OpenVMS Boot Flag Settings .....	6-13

7-1	Summary of SRM Commands .....	7-2
7-2	Notation Formats for SRM Console Commands .....	7-4
7-3	Special Characters for SRM Console .....	7-5
7-4	Device Naming Conventions .....	7-17
7-5	SRM Environment Variables for Soft Partitions .....	7-23
7-6	SCM Environment Variables for Hard Partitions .....	7-26
8-1	SCM Commands.....	8-14
8-2	Status Command Fields .....	8-18
8-3	Elements of Dial String and Alert String .....	8-28
8-4	SCM Troubleshooting.....	8-30
A-1	PCI Backplane Jumpers .....	A-1
A-2	HPM Jumpers .....	A-2
A-3	Standard I/O Module Jumpers.....	A-2



# Preface

## Intended Audience

This manual is for managers and operators of *Compaq AlphaServer 80/160/320* family systems.

## Document Structure

This manual uses a structured documentation design. Topics are organized into small sections, usually consisting of two facing pages. Most topics begin with an abstract that provides an overview of the section, followed by an illustration or example. The facing page contains descriptions, procedures, and syntax definitions.

This manual has eight chapters, an appendix, and glossary.

- **Chapter 1, Introduction**, gives a general overview of the *AlphaServer 80/160/320* family.
- **Chapter 2, GS320 System Overview**, describes the components of the GS320 system.
- **Chapter 3, GS320 System Configuration Rules**, provides configuration guidelines for the GS320 system.
- **Chapter 4, GS80 Rack System Overview**, describes the components of the GS80 rack system.
- **Chapter 5, GS80 Rack System Configuration Rules**, provides configuration guidelines for the GS80 system.
- **Chapter 6, Booting and Installing an Operating System**, tells how to boot a supported operating system and how to set boot options.
- **Chapter 7, Operation**, gives basic operating instructions.
- **Chapter 8, Using the System Control Manager**, describes the function and operation of the system control manager.

- **Appendix A, Jumpering Information**, calls out jumpers and their functions.

## Documentation Titles

**Table 1 AlphaServer 80/160/320 Family Documentation**

Order Number	Title
QA-6GAAA-G8	AlphaServer GS80/160/320 Documentation Kit
EK-GS320-UG	<i>AlphaServer GS80/160/320 User's Guide</i>
EK-GSCON-IN	<i>AlphaServer GS80/160/320 System Management Console Installation and User's Guide</i>
EK-GS320-RM	<i>AlphaServer GS80/160/320 Firmware Reference Manual</i>
EK-GSPAR-RM	<i>AlphaServer GS80/160/320 Getting Started with Partitions</i>
EK-GS320-IN	<i>AlphaServer GS160/320 Installation Guide</i>
EK-GSR80-IN	<i>AlphaServer GS80 Installation Guide</i>
AG-RKSWB-BE	AlphaServer GS80/160/320 User Information CD (HTML files)
AG-RLVJA-BE	AlphaServer GS80/160/320 User Information CD (translations)
QA-6GAAB-G8	AlphaServer GS80/160/320 Service Documentation Kit
EK-GS320-SV	<i>AlphaServer GS80/160/320 Service Manual</i>
EK-GS320-RM	<i>AlphaServer GS80/160/320 Firmware Reference Manual</i>
AG-RKSZA-BE	AlphaServer GS80/160/320 Service Information CD
EK-GS320-UP	<i>AlphaServer GS160/320 Upgrade Manual</i>
EK-GSR80-UP	<i>AlphaServer GS80 Upgrade Manual</i>
EK-GS320-SP	<i>AlphaServer GS80/160/320 Site Preparation</i>

## Information on the Internet

Visit the Compaq Web site at [www.compaq.com](http://www.compaq.com) for service tools and more information about the *AlphaServer* 80/160/320 family systems.

# Chapter 1

## Introduction

The *Compaq AlphaServer* GS160/320 and GS80 systems are high-performance server platforms designed for enterprise-level applications. They offer a high degree of scalability and expandability.

The GS160/320 system uses up to four Alpha microprocessors in each quad building block (QBB). Two QBBs are paired back-to-back and rotated 180 degrees with reference to each other and then enclosed in a system box. A system cabinet can hold up to two system boxes.

The GS80 rack system uses up to four Alpha microprocessors in a drawer. Each rack holds up to two system drawers.

This chapter introduces the *AlphaServer* GS160/320 and *AlphaServer* GS80 systems. There are three sections:

- *AlphaServer* GS160/320 and GS80 Systems
- Firmware and Utilities Overview
- System Architecture

---

**NOTE:** *When you unpack your system, be sure to save and store all shipping brackets, pallets, and packing material. You will need this material to repack the system, if you should decide to relocate it.*

---

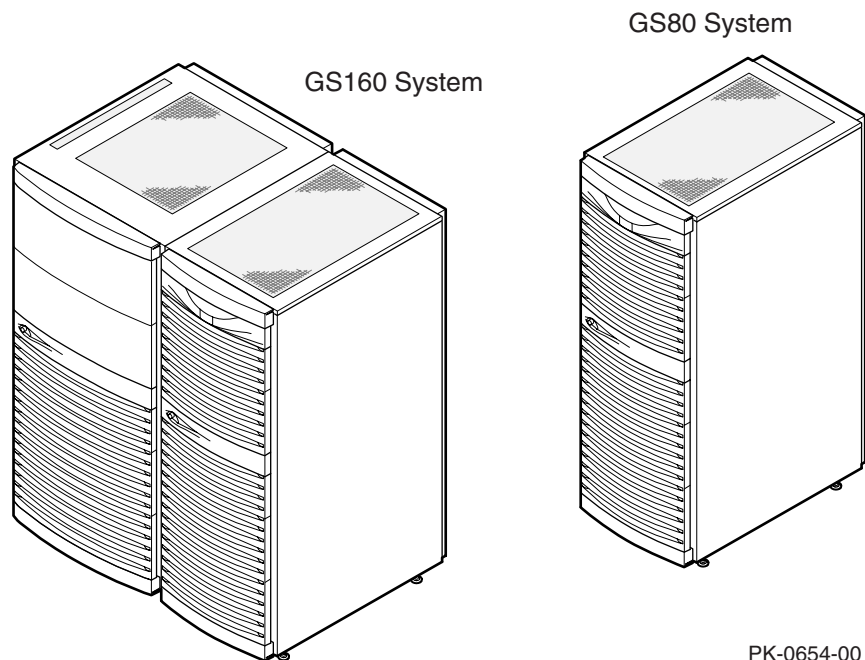
## 1.1 AlphaServer GS160/320 and GS80 Systems

---

The AlphaServer GS160/320 system and GS80 rack system are separate, but related, in that they use the same switch technology. The CPU modules, memory modules, and power modules are also the same. In the GS160/320 system, the modules are in a system box in a cabinet. In the GS80 rack system, the modules are in a drawer.

---

Figure 1-1 AlphaServer GS160 and GS80 Systems



### **AlphaServer GS160/320 System**

The *AlphaServer* GS160 system cabinet contains up to two system boxes supporting a maximum of 16 CPU modules.

In an *AlphaServer* GS320 system, a second system cabinet is used to expand the system (up to four system boxes containing a maximum of 32 CPU modules).

A power cabinet contains the power components, I/O boxes, and storage. Additional I/O and storage can be housed in expander cabinets.

### **AlphaServer GS80 System**

The *AlphaServer* GS80 rack system contains up to two system drawers, I/O, and storage. Each system drawer supports up to four CPU modules, for a maximum of eight CPUs per system. Power components are mounted at the bottom of the rack cabinet.

An expander cabinet can be used to house additional I/O and storage.

### **System Management Console**

The console device, called the system management console (SMC), is a *Compaq Deskpro* PC, a DECserver 90M terminal server, and associated hardware and software. For installation instructions and user information, see the *AlphaServer GS80/160/320 System Management Console Installation and User's Guide*.

## 1.2 Firmware and Utilities Overview

---

**Firmware residing in ROM on CPU and other modules in the system provides commands for booting the operating system, testing devices and I/O adapters, and other tasks useful in operating and maintaining a running system. You type commands at the console device.**

---

### SRM Console

Systems running the *Tru64 UNIX* or *OpenVMS* operating systems are configured from the SRM console, a command-line interface (CLI). From the CLI you can enter commands to configure the system, view the system configuration, and boot the system.

### System Control Manager (SCM)

The SCM firmware allows the user to access the system remotely. In addition, the SCM:

- Monitors and notifies the user of power or temperature alert conditions
- Controls initial system power-up
- Manages hard partitions

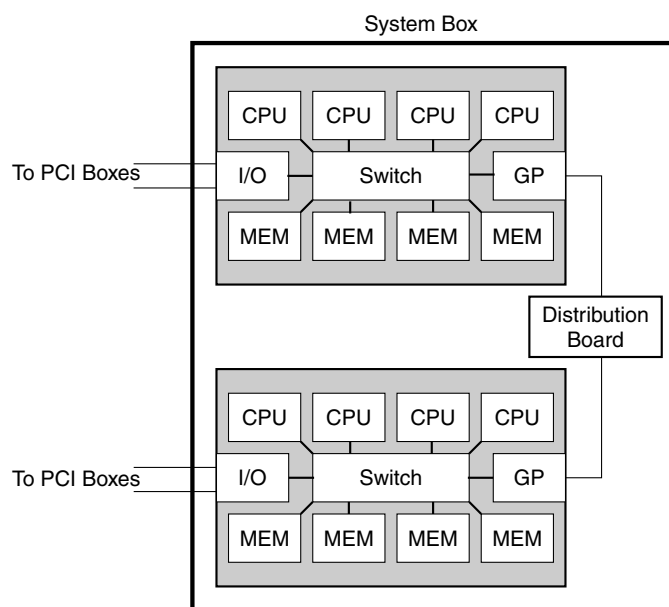
### LFU (Loadable Firmware Update Utility)

You can boot this utility (with the SRM **boot** command) whenever you need to update the SRM console firmware or I/O device firmware. The CD with the *AlphaServer* firmware is updated periodically.

## 1.3 System Architecture

Each QBB in a GS160/320 system and each QBB (system drawer) in a GS80 system has a backplane and a switch supporting the CPU modules, memory modules, and I/O riser modules. Figure 1-2 shows two QBBs in a single-box system.

Figure 1-2 Sample System Architecture



PK-0601A-98

The backplane of each QBB (GS160/320 systems) and drawer (GS80 rack systems) contains the switch that interconnects the CPU modules, memory modules, and I/O riser modules.

The global port and distribution board provide the interconnect for the QBBs. In a two-box system, a hierarchical switch is used as the interconnect, in place of the distribution board.

I/O riser modules connect the PCI boxes to the QBB backplane.





# Chapter 2

## GS160/320 System Overview

Each system cabinet contains one or two system boxes. The system box houses two quad building blocks, or QBBs. CPU modules, memory modules, power modules, and I/O riser modules plug into the QBB backplane.

The power cabinet contains power components, PCI boxes, and storage shelves.

This chapter provides an overview of the system in these sections:

- System Characteristics
- System Box Architecture
- Quad Building Block (QBB) Components
- Power System
- PCI I/O
- Control Panel

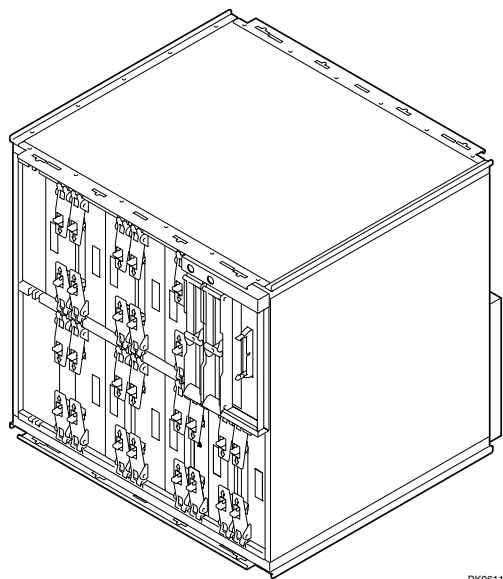
## 2.1 System Characteristics

---

The illustration shows the BA51A-AA system box. Table 2-1 lists the system box characteristics. Table 2-2 lists power cabinet and environmental characteristics.

---

**Figure 2-1 System Box**



PK0611

**Table 2-1 System Box Characteristics**

Characteristic	Specification
Size	535 mm H x 550 mm W x 475 mm D (21.06 in. x 21.65 in. x 18.7 in.)
Weight	54.55 kg (120 lb) maximum
Maximum heat dissipation	2000 w (6,850 Btu/hr)

**Table 2-2 Power Cabinet and System Environmental Characteristics**

Power Cabinet	Specifications
<b>Electrical</b>	
Voltage	120/208 VAC (U.S) 380–415 VAC (Europe) 200 VAC (Japan)
Phase	3-phase
Frequency	50–60 Hz
Maximum input current/phase	21 A
Heat dissipation <sup>1</sup>	9,300 W/31,800 Btu/Hr
<b>Environmental</b>	
Temperature	Operating: 5° to 35°C (41 to 95°F) Not operating: -40° to 66°C (-40° to 151°F)
Humidity	Operating: 10% to 90% Not operating: 10% to 95%
Altitude	Operating: 0 to 3 km (0 to 10,000 ft) Not operating: 0 to 12.2 km (0 to 40,000 ft)

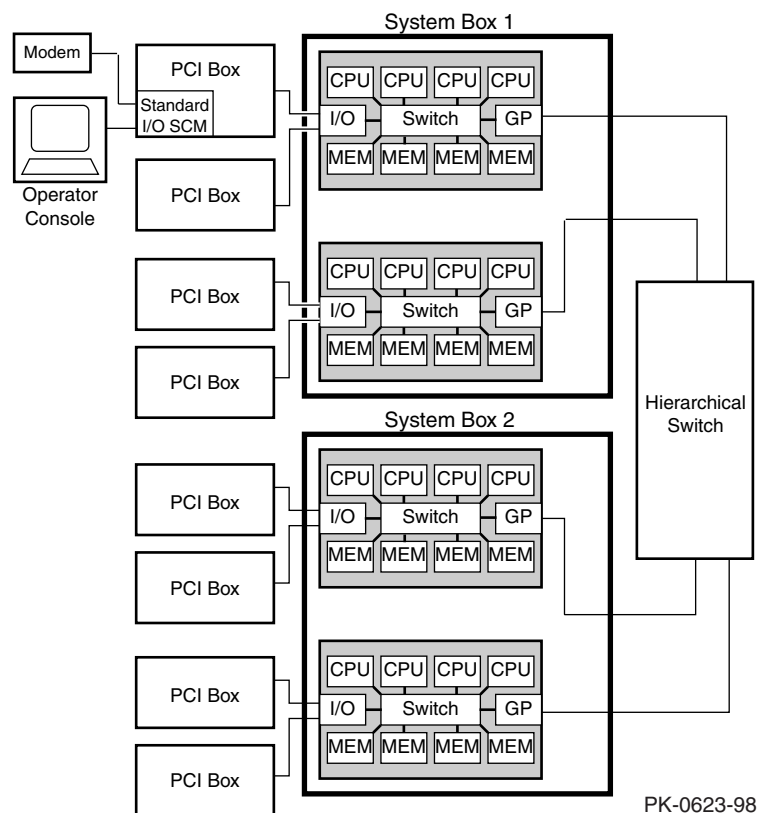
---

<sup>1</sup> A fully configured system with three system boxes, nine power supplies, two PCI boxes, and storage shelf.

## 2.2 System Box Architecture

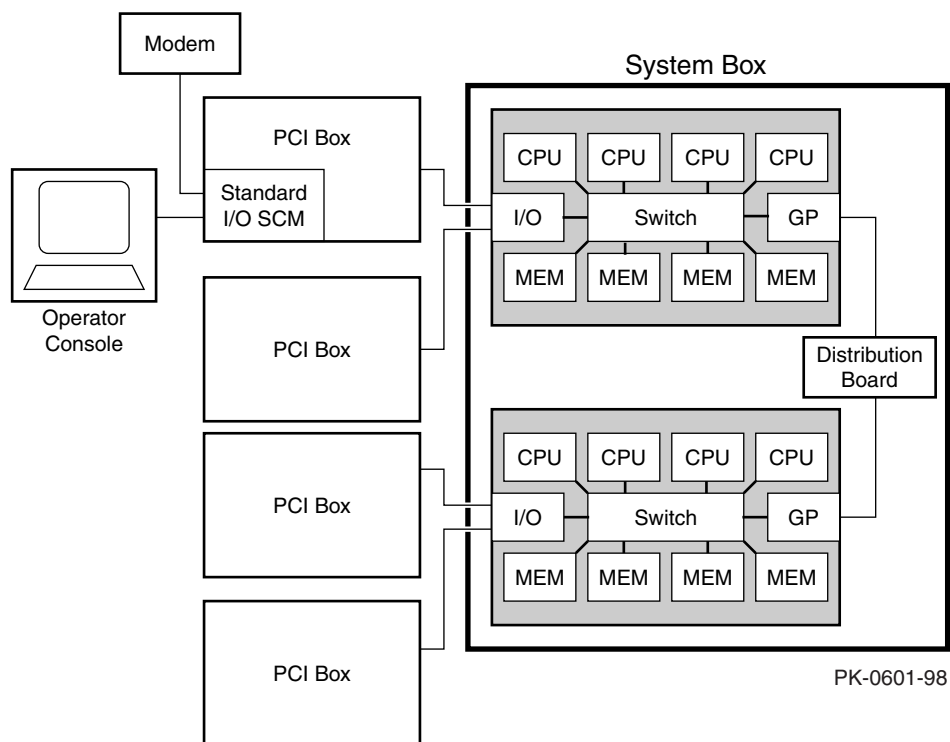
The system box houses two quad building blocks (QBBs). Each QBB has a backplane with a switch interconnect that supports up to four CPU modules, four memory modules, two power modules, two I/O riser modules, and a global port. Figure 2-2 shows two system boxes connected by the hierarchical switch. Figure 2-3 shows one system box and the distribution board.

Figure 2-2 System Box Block Diagram (16-Processor System)



The switch on the backplane connects the CPU modules, memory modules, I/O riser modules, and global port. In an 8-P system, the global ports connect the QBBs to the distribution board. In a 16-P or a 32-P system, the global ports connect the QBBs to the hierarchical switch.

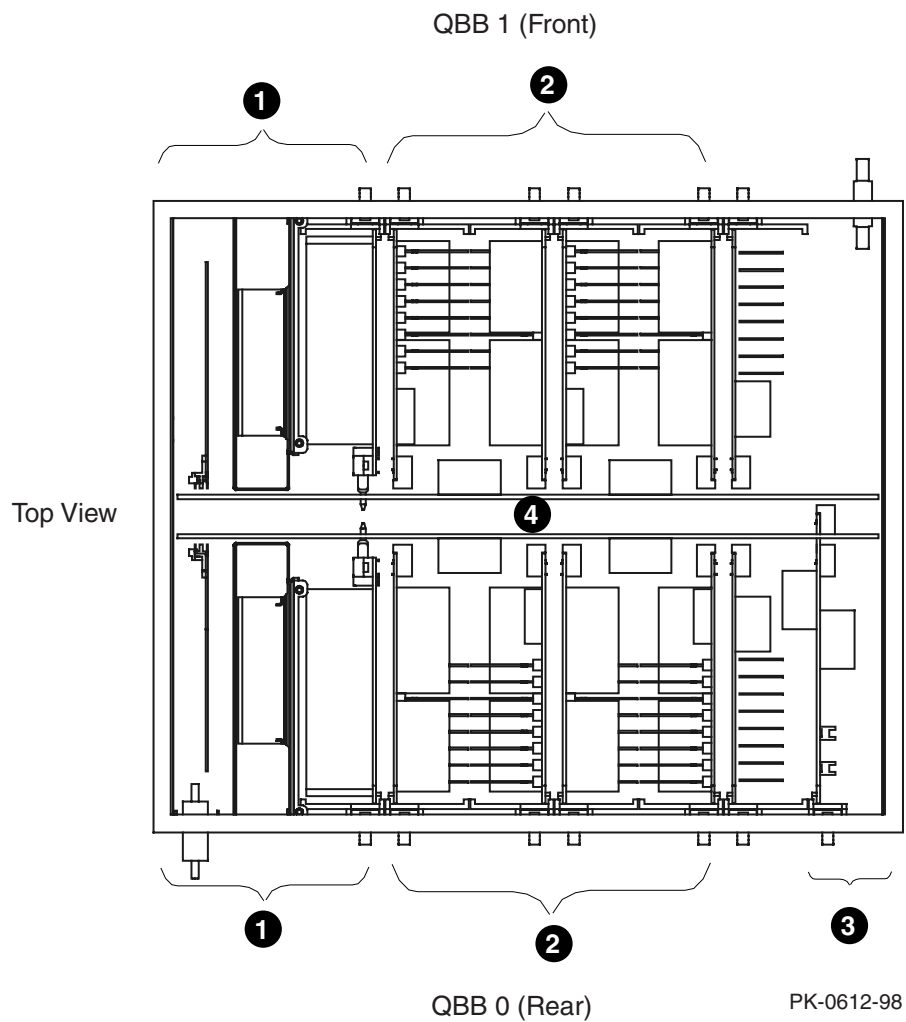
### Figure 2-3 System Box Block Diagram (8-Processor System)



## 2.3 Quad Building Block (QBB) Components

Figure 2-4 shows two QBBs back to back in the system box.

Figure 2-4 System Box QBBs (Top View)



The QBB backplanes are attached to a stiffener and mounted in a system box enclosure. Each backplane has a differently positioned cutout to accommodate the global port modules. A global port module is mounted on the front of one QBB and the other is mounted on the back of the other QBB, putting both global port modules near the distribution board (or the hierarchical switch) when the system box is installed in the cabinet.

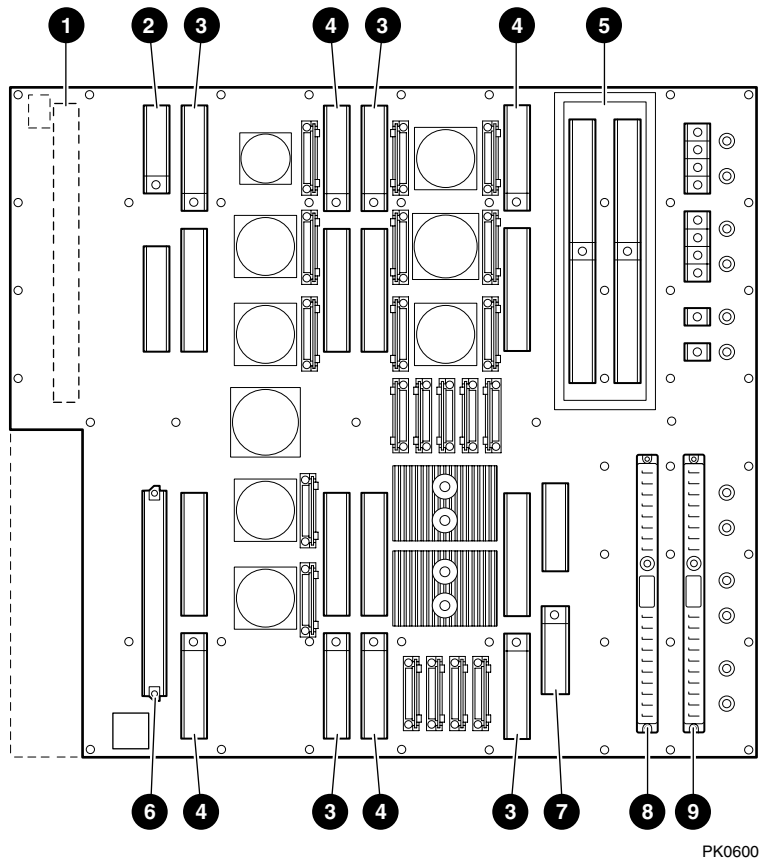
The callouts in Figure 2–4 point to the location of these components in the system box:

- ❶ I/O riser module area. Each QBB supports up to two I/O riser modules.
- ❷ CPU module and memory module area. Each QBB holds four CPUs and four memory modules.
- ❸ Global port area.
- ❹ Backplane area.

### 2.3.1 Backplane

Each QBB backplane is located at the center of the system box. Figure 2-5 shows an unpopulated backplane (no modules installed) as you would see it from the front of the system box.

Figure 2-5 Backplane (System Box, Front View)





The CPU, memory, power, and I/O riser modules plug into the backplane. Because of the orientation of the QBB backplanes, the modules are situated differently in the front and rear of the system box. See Section 3.5 for more information.

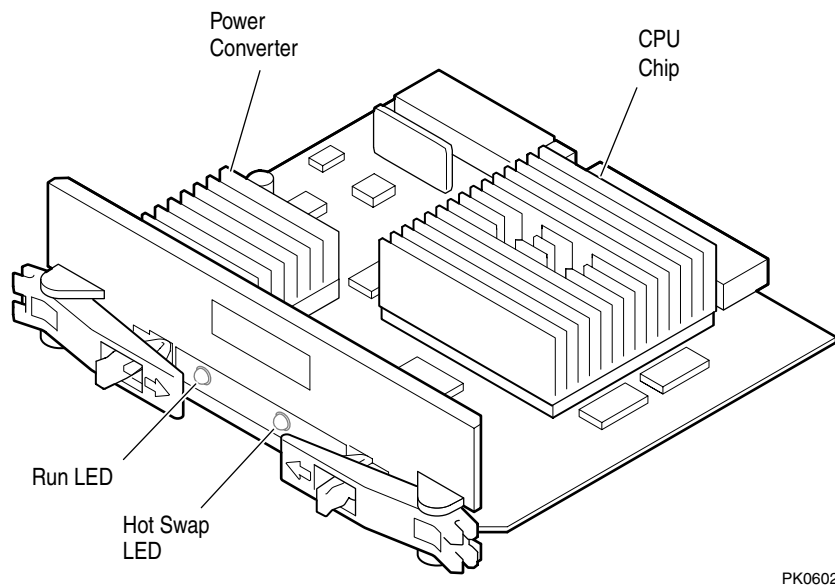
The switch interconnect on the backplane allows any processor to access any memory on the QBB. The backplane also provides power to the modules.

- ❶ Global port module
- ❷ Directory module
- ❸ CPU module
- ❹ Memory module
- ❺ I/O riser modules
- ❻ Clock splitter module
- ❼ Power system manager (PSM) module
- ❽ Main power module
- ❾ Auxiliary power module

### 2.3.2 CPU Module

A CPU module comes with an Alpha microprocessor chip with a B-cache, cache control and TAG comparison logic, clock logic, and a DC-DC power converter. Also included on the module is logic for implementing self-test diagnostics. Each module has a Run LED and a Hot Swap LED.

Figure 2-6 CPU Module

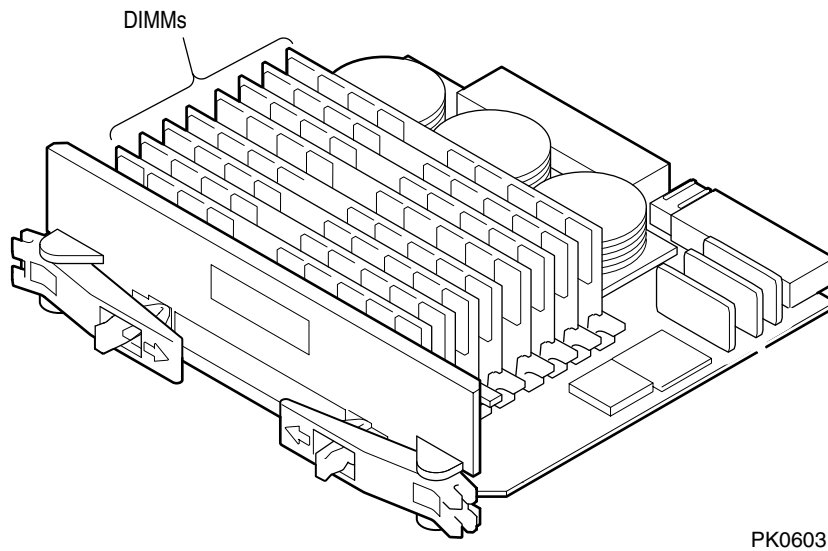


PK0602

### 2.3.3 Memory Module

A memory module has eight DIMM slots. See Section 3.7 for memory configuration guidelines.

Figure 2-7 Memory Module

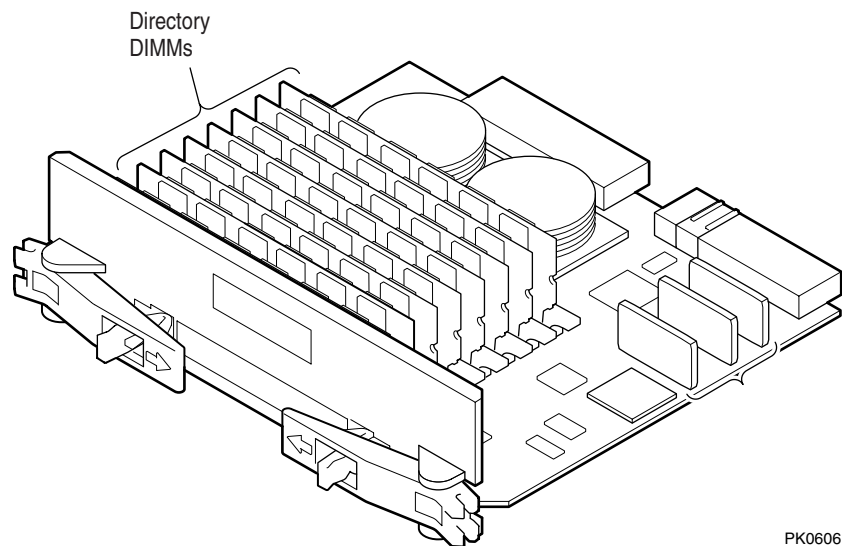


PK0603

### 2.3.4 Directory Module

In a GS160/320 system, one directory module is required for each QBB in a system box. In a two-drawer GS80 system, a directory module is required in each system drawer. No directory module is needed in a one-drawer system. The directory module functions as a memory coherency manager.

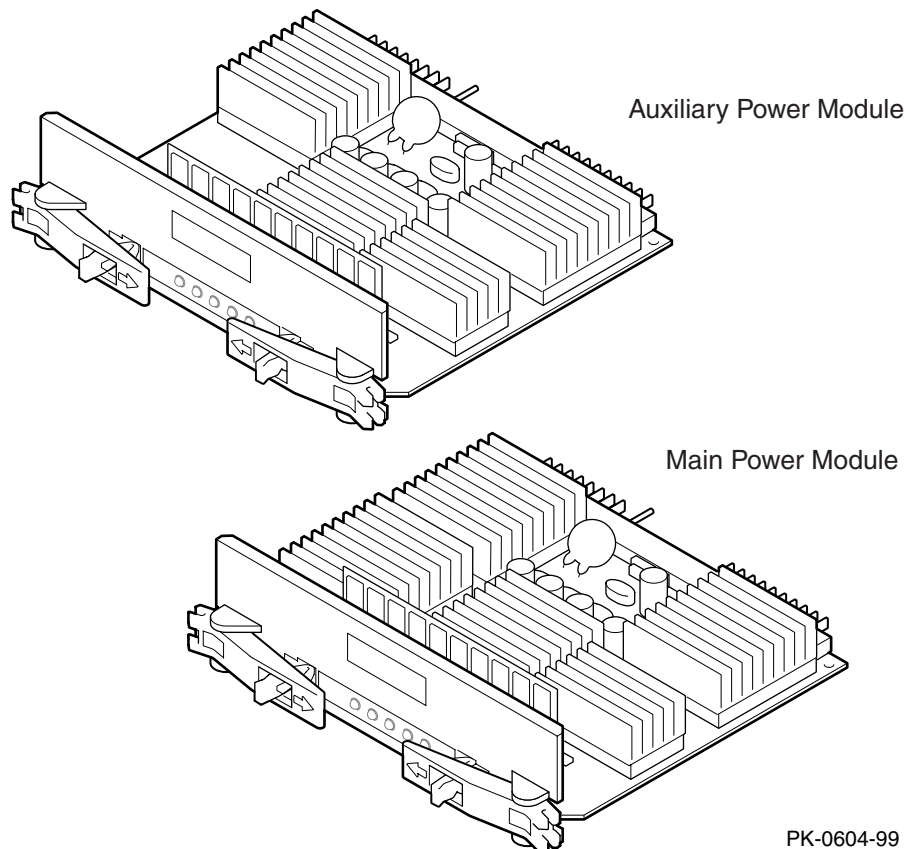
Figure 2-8 Directory Module



### 2.3.5 Power Modules

Two power modules are installed in the QBB backplane. The main power module and the auxiliary power module convert 48 VDC to the various voltages required to power the QBB.

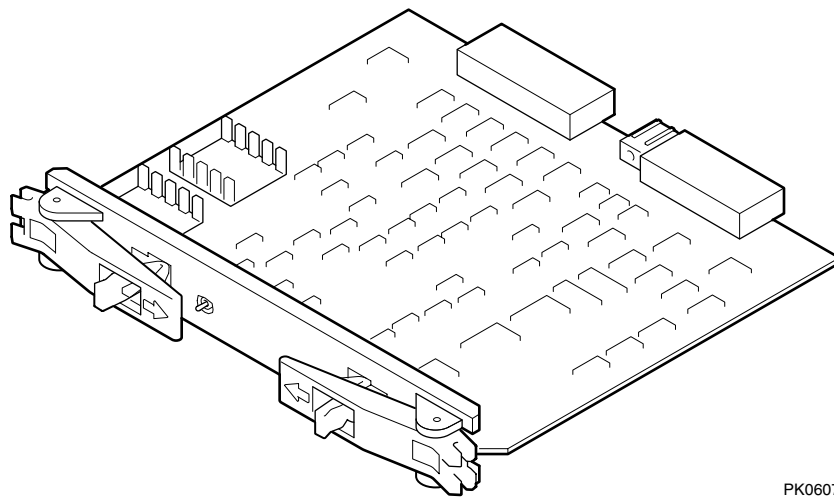
Figure 2-9 Power Modules



### 2.3.6 Power System Manager Module

Each QBB has one power system manager (PSM) module. This module monitors CPUs, voltages, temperatures, and blower speed in the cabinet and reports this information to the system control manager (SCM).

Figure 2-10 Power System Manager Module



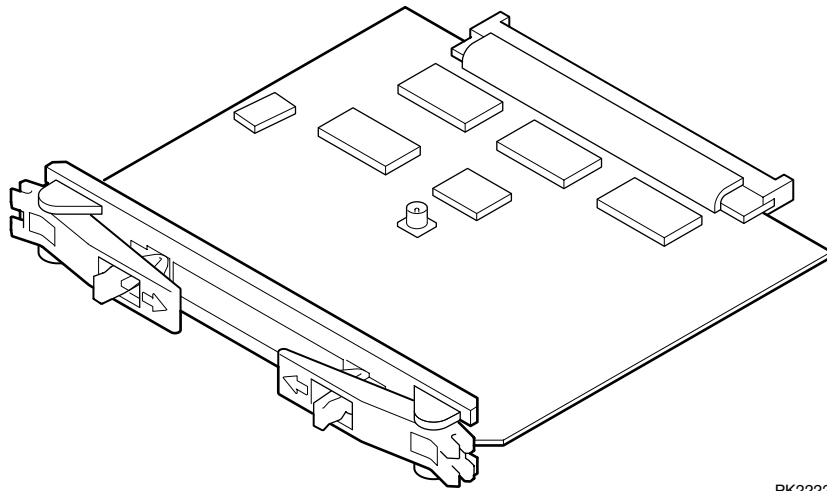
PK0607

The PSM module is connected to other PSM modules and the SCM micro-processor (located on the standard I/O module) through the console serial bus (CSB). The SCM is the master; the PSM can only operate as a slave. The PSM controls the powering on/off and resetting of all modules within the QBB and of the QBB itself. The PSM also contains three I<sup>2</sup>C bus interfaces and a serial I/O bus channel to each CPU in the QBB. The PSM module retrieves information from the I<sup>2</sup>C EEROMs residing on the CPUs, memory modules, and the QBB backplane. The PSM uses a serial I/O port connection to communicate with one CPU module at a time. The PSM can only perform a function as the result of a request from the SCM.

### 2.3.7 Clock Splitter Module

The clock splitter module converts one global signal to identical copies of a signal that is then distributed to master phase lock loops associated with the ASICs and the system processors within a QBB. It also generates independent clock signals for the I/O domain.

Figure 2-11 Clock Splitter Module



PK2222

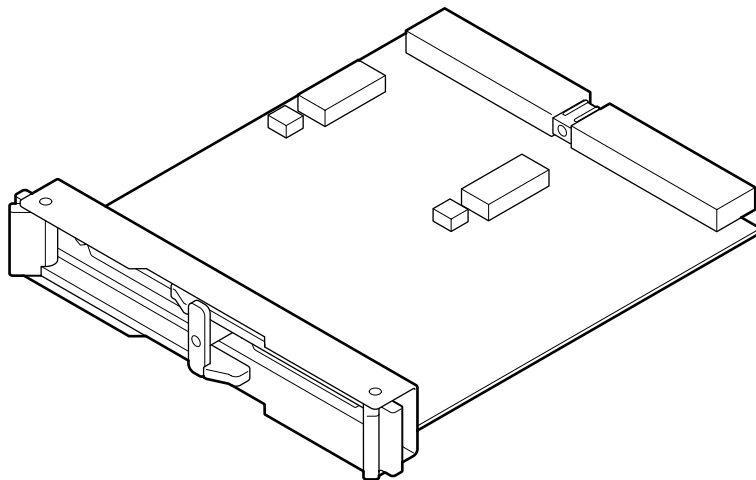
### 2.3.8 I/O Riser Module

---

The I/O riser module is used to connect the QBB backplane to a PCI box. A “local” I/O riser module is located on the QBB backplane; a “remote” I/O riser module is in the PCI box.

---

Figure 2-12 I/O Riser Module



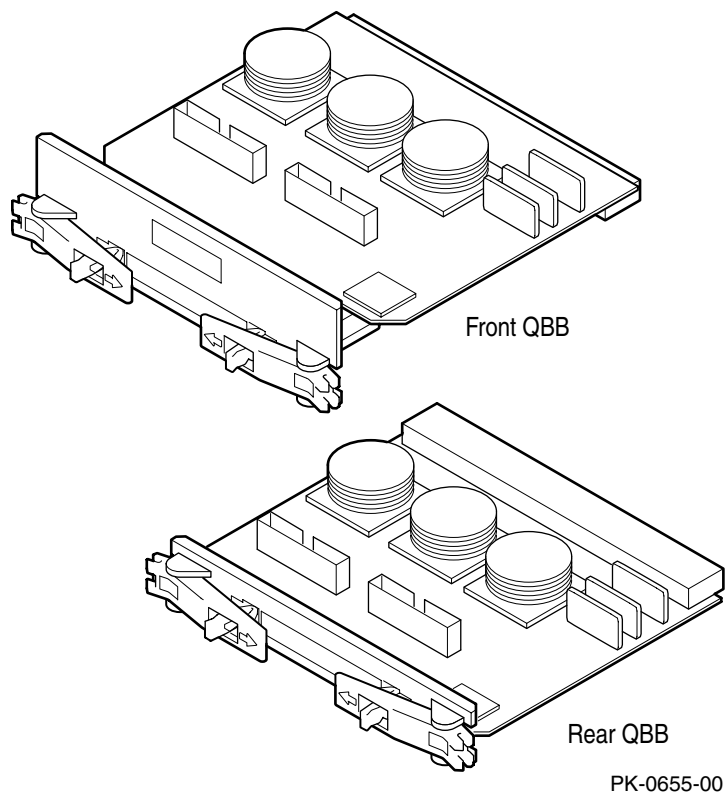
PK0605



### 2.3.9 Global Port Module

The global port provides the interconnect to the other QBB(s) through the distribution board or the hierarchical switch.

Figure 2-13 Global Port Module



### 2.3.10 Distribution Board

In single-box systems, a distribution board connects the two QBBs through the global ports.

Figure 2-14 Distribution Board

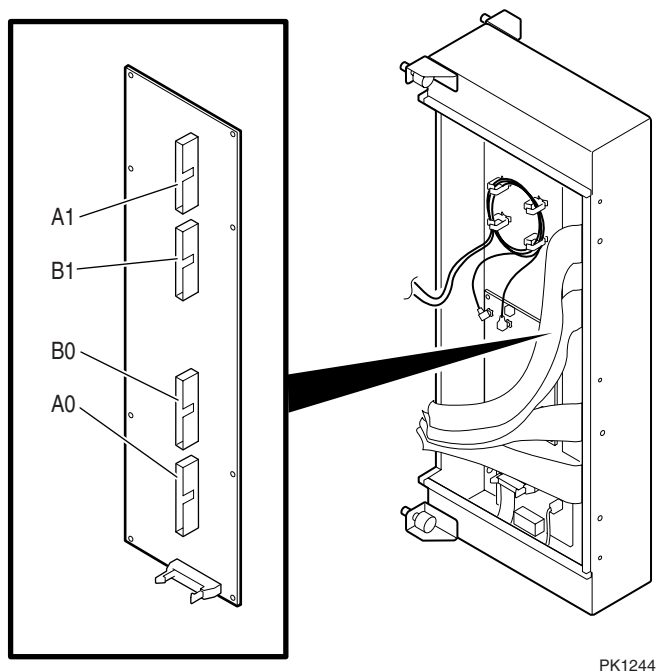
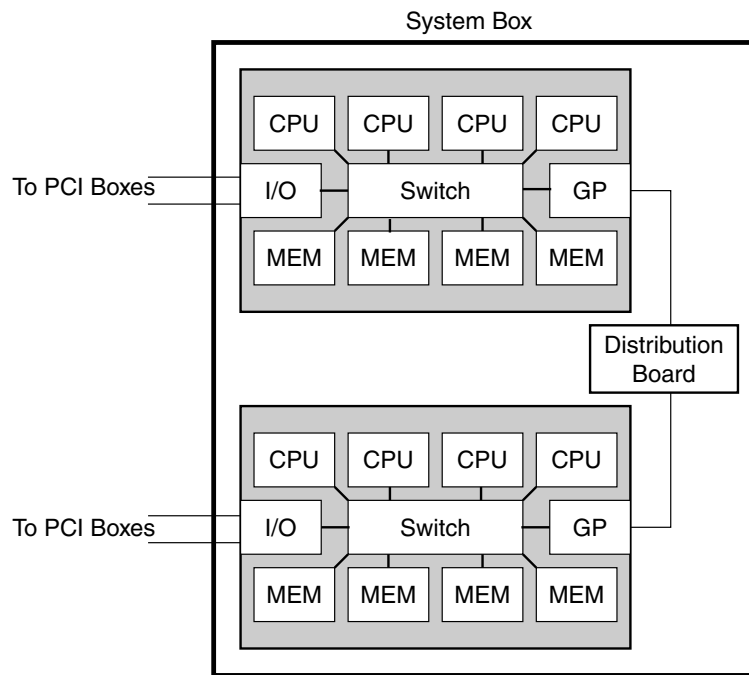


Figure 2–15 is a block diagram showing the distribution board as the interconnect between two QBBs.

**Figure 2–15 Distribution Board in Single-Box System**

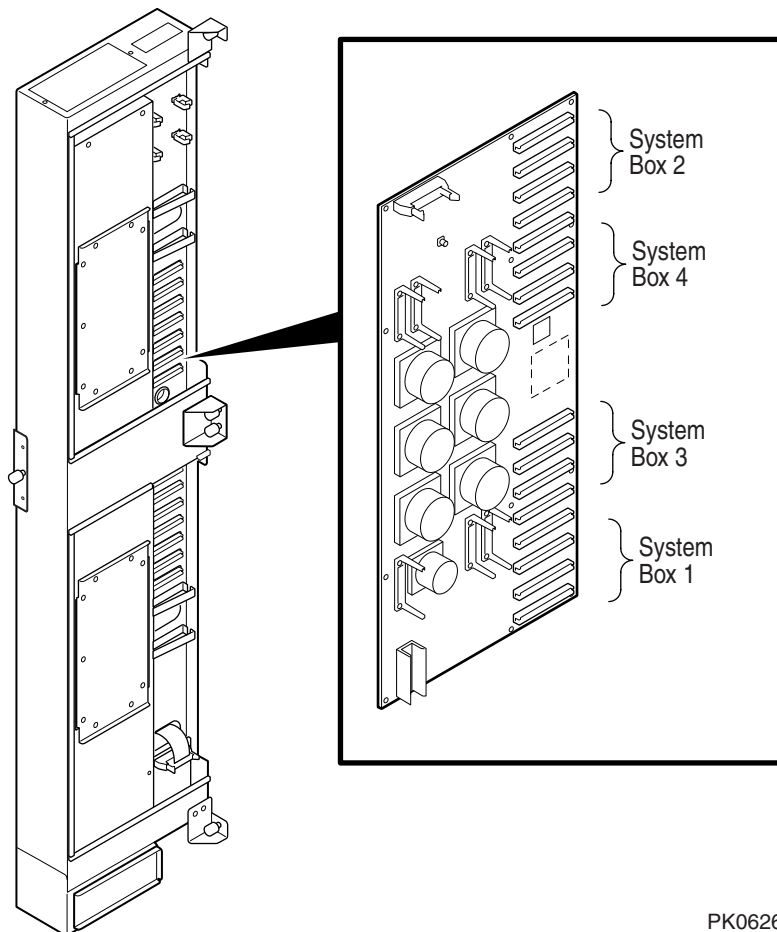


PK-0601A-98

### 2.3.11 Hierarchical Switch

In two-box systems, a hierarchical switch links the QBBs through the global ports. The hierarchical switch connects QBBs in three- and four-box systems also.

Figure 2-16 Hierarchical Switch



PK0626

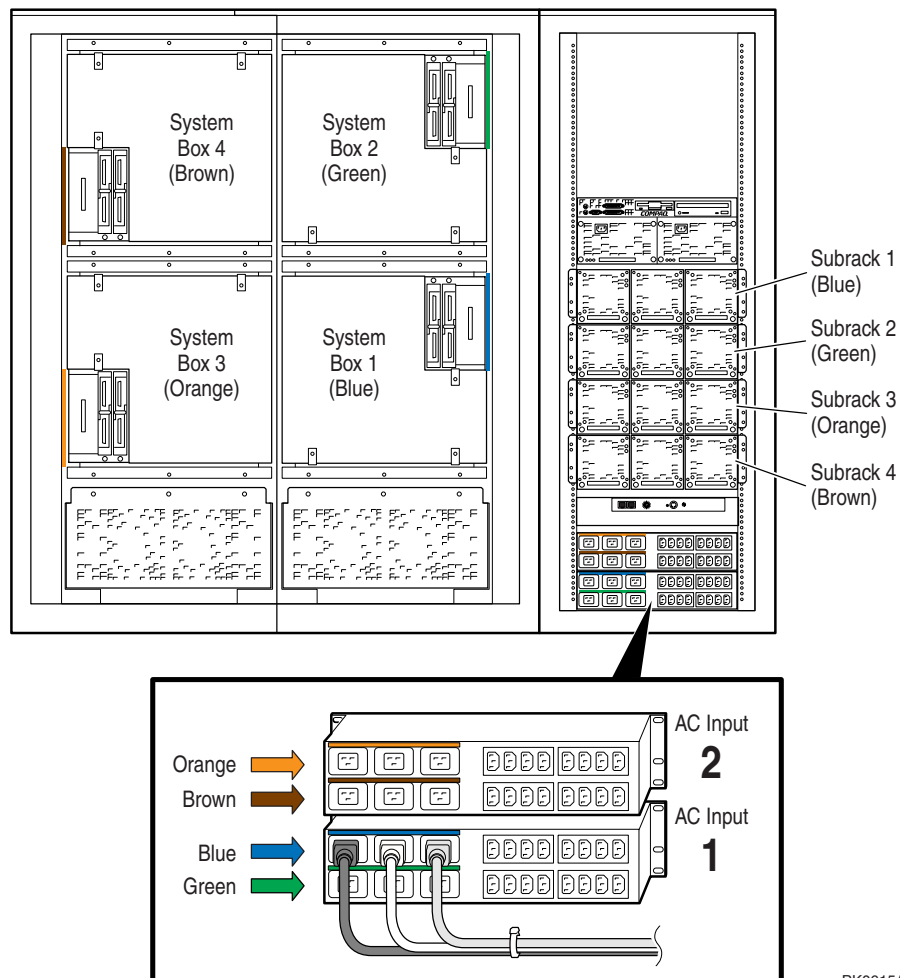
The hierarchical switch links the QBBs in systems having more than one system box. Figure 2–16 shows cable connectors for each system box (a pair of connectors for two signal cables routed to each QBB global port in the system).

The hierarchical switch power manager (HPM) module controls power and monitors the temperature inside the hierarchical switch housing. The HPM module, along with the PSM modules and PBM modules, report status information to the SCM.

## 2.4 Power System

Each system box has a power subrack with up to three 48 VDC power supplies. Figure 2-17 shows the power system for a 32-P system. See Section 3.3 for power configuration rules.

Figure 2-17 Power System



PK0615A

Power cables and components are color-coded to ensure proper identification and easy handling.

---

**NOTE:** *Color-coded components and power cables must match to ensure proper power distribution, particularly in hard-partitioned systems.*

---

Figure 2–17 shows each system box and its color-related power subrack and AC input box. The AC input box also has color-coded circuit breakers. Each AC input box provides power to the subracks, PCI boxes, and storage shelves.

Power distribution and signal cables are connected from the power subrack to the QBB backplanes through a power distribution bulkhead. Another cable is used to provide power to the blower, located at the bottom of the system cabinet, and the control panel, located in the power cabinet.

Each QBB has its own main power module and auxiliary power module. Each CPU module has its own power converter that converts the 48 VDC to the required voltage.

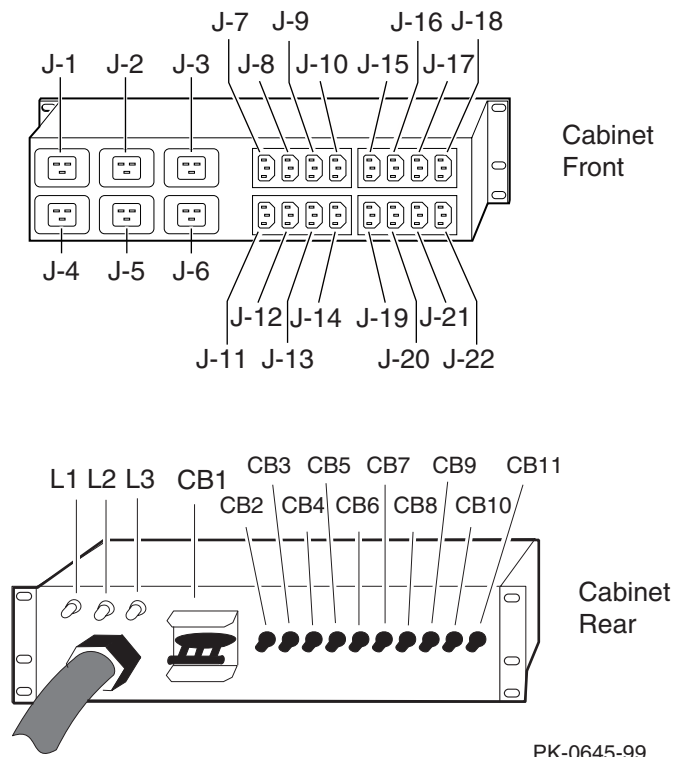
When the main power circuit breaker (CB1) is on and the AC input box is plugged in, the console serial bus (CSB) has auxiliary power (Vaux), enabling the system control manager (SCM) to power up the system.

See Section 2.4.1 for more information on the AC input box and circuit breakers.

## 2.4.1 AC Input Box

A system has two AC input boxes. Figure 2-18 shows the circuit breakers (CB1–CB11), LEDs (L1–L3), and connectors (J1–J22) on the AC input box.

Figure 2-18 AC Input Box





The three LEDs on the AC input box should be lit at all times, indicating that all three power phases are present in the 3-phase AC input.

Table 2–3 lists the AC input box circuit breakers and the lines they protect.

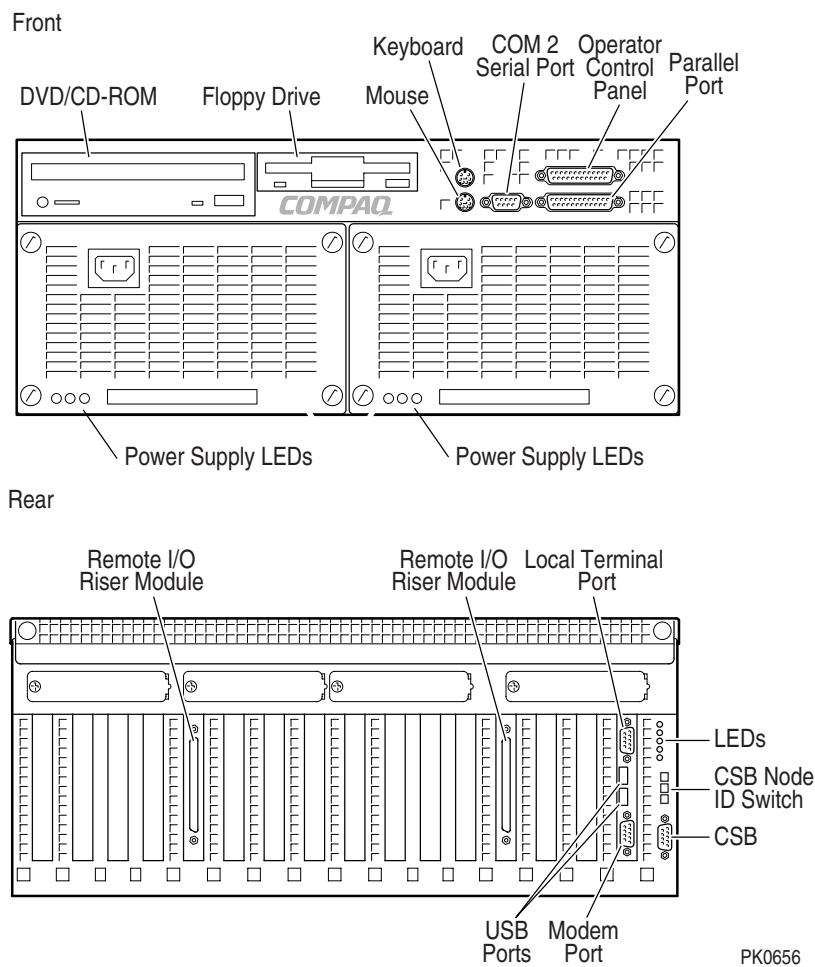
**Table 2–3 AC Input Box Circuit Breakers**

<b>Circuit Breaker</b>	<b>Line(s) Protected</b>
CB1 (Main)	All lines protected.
CB2	J1
CB3	J2
CB4	J3
CB5	J4
CB6	J5
CB7	J6
CB8	J7, J8, J9, J10
CB9	J11, J12, J13, J14
CB10	J15, J16, J17, J18
CB11	J19, J20, J21, J22

## 2.5 PCI I/O

The power cabinet contains at least one PCI master box, and may contain PCI expansion boxes.

Figure 2-19 PCI Master Box



A PCI master box has a standard I/O module, a DVD/CD-ROM drive and a floppy drive as shown in Figure 2–19. PCI expansion boxes provide additional slots for options. Each PCI power supply has three LEDs: Vaux OK, Power OK, and Swap OK.

#### **BA54A-AA PCI Box**

The BA54A-AA PCI box is a PCI master box. It contains the following components:

- Backplane with a standard I/O module
- Thirteen I/O option slots
- DVD/CD-ROM drive
- SCSI disk drive
- Floppy drive
- Ports: one local terminal port, one serial port, one modem port, one parallel port, two USB ports, one keyboard port, one mouse port, one CSB port, and one control panel port
- Two I/O riser slots
- Two power supplies

#### **BA54A-BA PCI Box**

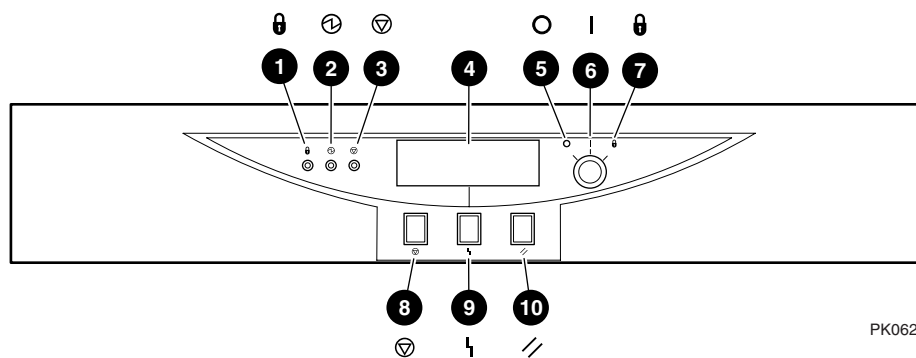
The BA54A-BA box is a PCI expansion box and contains the following components:

- Backplane with fourteen option slots
- Two I/O riser slots
- Two power supplies

## 2.6 Control Panel

The control panel is located at the top of the power cabinet. It has a three-position **Off/On/Secure** switch, three pushbuttons, three status LEDs, and a diagnostic display.

Figure 2-20 Control Panel



PK0621


















The callouts in Figure 2–20 point to these components on the control panel:

- ❶ Secure LED – When lit, indicates that the keyswitch is in the Secure position and system is powered on. All pushbuttons and SCM functions are disabled, including remote access to the system.
- ❷ Power OK LED – When lit, indicates at least one QBB is powered on and remote console operations are enabled. The keyswitch is in the On position.
- ❸ Halt LED – When lit with the Power LED, indicates the system is powered on, remote console operations are enabled, and the Halt pushbutton is pressed.  
  
When lit with both Power LED and Secure LED, indicates a powered on system, disabled remote console operations, and Halt pushbutton is pressed in.
- ❹ Diagnostic display indicates system status.
- ❺ Off switch position – System is powered off and cannot be powered on remotely.
- ❻ On switch position – System is enabled to be powered up. A remote system user can power on or power off the system.
- ❼ Secure switch position – System is powered on and cannot be remotely powered on or off. All pushbuttons and SCM functions are disabled.
- ❽ Halt pushbutton – Pressing this pushbutton causes the operating system to perform a halt, with no captured error information. The system will automatically reset if the **auto\_fault\_reset** environment variable is enabled.
- ❾ Fault pushbutton – Pressing this pushbutton causes a system reset without clearing captured error information in the control and status registers.
- ❿ Reset pushbutton – Pressing this pushbutton causes a system reset that clears captured error information.

2.6.1 Control Panel LEDs

Figure 2-21 shows the various control panel LED status indications.

Figure 2-21 Control Panel LED Status

Control Panel LEDs			Status
Secure	Power OK	Halt	
			System powered on; remote console disabled; pause mode.
			System powered on; remote console disabled.
			System powered on; remote console enabled; remote console halt or Halt button depressed.
			System powered on; remote console enabled.
			System powered off for any of the following reasons: <ul style="list-style-type: none"><li>o No AC power available</li><li>o Keyswitch in Off position</li><li>o Keyswitch in On position but system powered off by remote console or power/temperature failure</li></ul>
	On		
	Off		

PK-0622-99

# Chapter 3

## GS160/320 System Configuration Rules

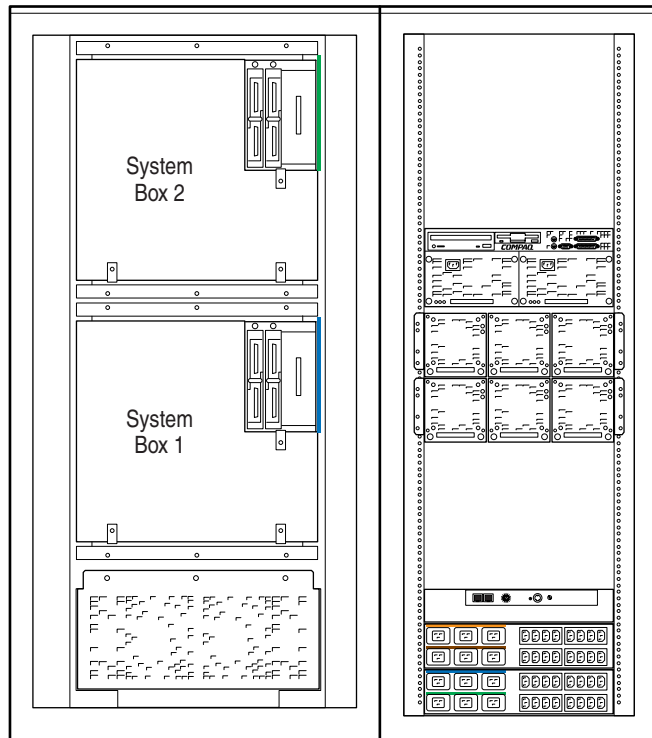
This chapter provides configuration rules for the following:

- GS160 System Cabinet
- GS320 System Cabinets
- Power Cabinet
- System Box
- QBB Color Code
- Memory Configurations
- Memory Interleaving Guidelines
- PCI Boxes
- PCI Box Slot Configuration
- Expander Cabinet

## 3.1 GS160 System Cabinet

Figure 3-1 shows the front view of the system cabinet and the power cabinet. One system cabinet houses either one system box or two system boxes. In a one-box system, a distribution board connects the two QBBs. In a two-box system, a hierarchical switch connects the two QBBs.

Figure 3-1 GS160 System



PK0614



### **About the System Cabinet**

The cabinet contains the following components:

- Vertical mounting rails
- Wrist strap for static discharge protection

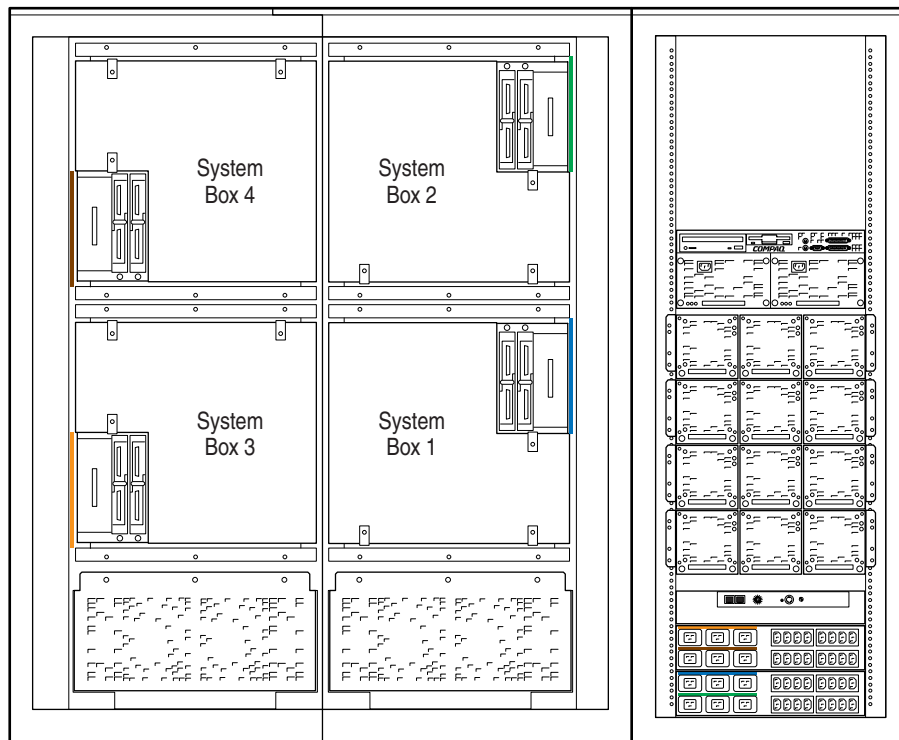
### **GS160 Configuration Rules**

- System box 1 (see Figure 3–1) is mounted in the lower half of the cabinet, above the blower.
- System box 2 is mounted in the upper half of the cabinet, over system box 1.

## 3.2 GS320 System Cabinets

Figure 3-2 shows the front view of the system cabinets. Two system cabinets house either three system boxes or four system boxes. A hierarchical switch is used to connect the QBBs.

Figure 3-2 GS320 System



PK0615

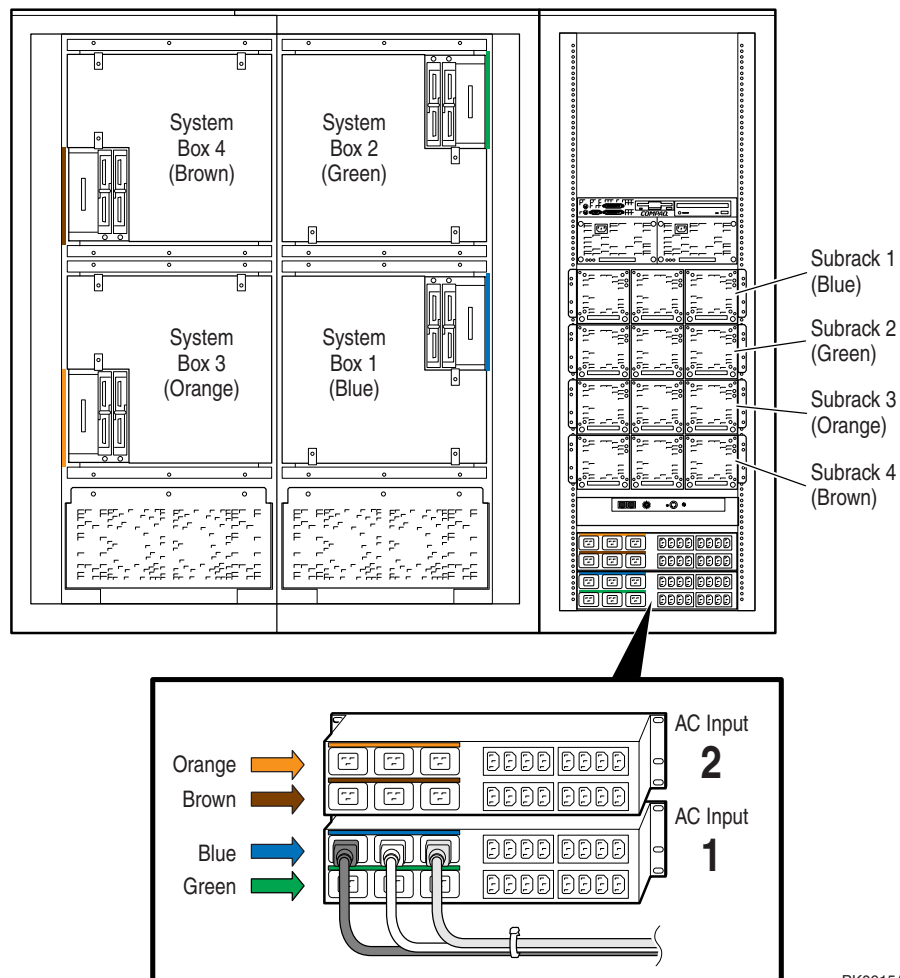
### **GS320 System Configuration Rules**

- In system cabinet 1, system box 1 (see Figure 3–2) is mounted in the lower half of the cabinet, above the blower. System box 2 is mounted in the upper half of the cabinet, above system box 1.
- In system cabinet 2, system box 3 is mounted in the lower half of the cabinet; system box 4, the upper half of the cabinet.

### 3.3 Power Cabinet

One power cabinet is required for all systems. The power cabinet houses the control panel, AC input boxes, power supplies, PCI I/O boxes, and storage.

Figure 3-3 Power Cabinet Configuration (32-P System)



PK0615A

### **Power System Requirements**

- Each system box requires a power subrack.
- Each power subrack has three power supplies. The third power supply is always redundant. See Section 3.3.1 for power supply slot assignments.
- Two AC input boxes are required.

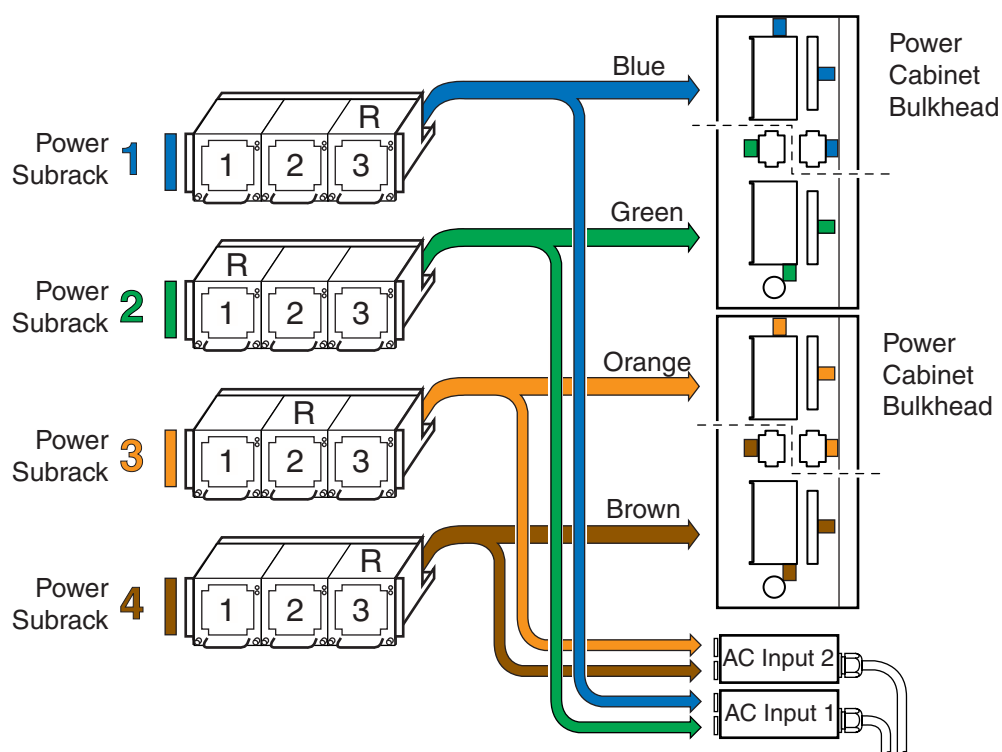
Cables, AC input boxes (including AC circuit breakers), power subracks, and system boxes are color-coded at cable connections to ensure proper cabling. Figure 3–3 shows the color coding scheme for a 32-P system.

PCI boxes and storage shelves are installed in the upper half of the power cabinet.

### 3.3.1 Power Supply Slot Assignments

Figure 3–4 show the power supply slot assignments in each power subrack.

Figure 3–4 Power Supply Slot Assignments



R indicates redundant power supply slot.

PK-0624-99

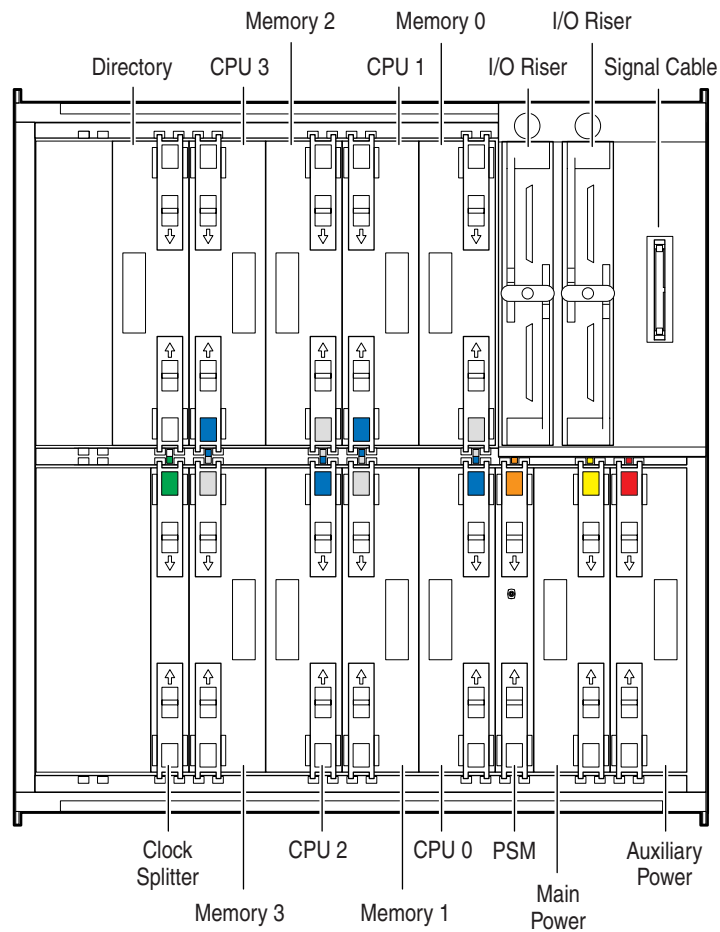
### **Power Supply Configuration Rules**

- Power subracks are always mounted in the same power cabinet location, regardless of the number of system boxes.
- Power supply slot assignments remain the same in all systems, regardless of the number of system boxes.
- A redundant power supply slot is always the last slot to be used in a subrack.

## 3.4 System Box

The system box contains two QBBs. Figure 3-5 shows a fully populated QBB as seen from the front of the cabinet. Figure 3-6 shows the second QBB at the rear of the cabinet.

Figure 3-5 System Box QBB (Cabinet Front)

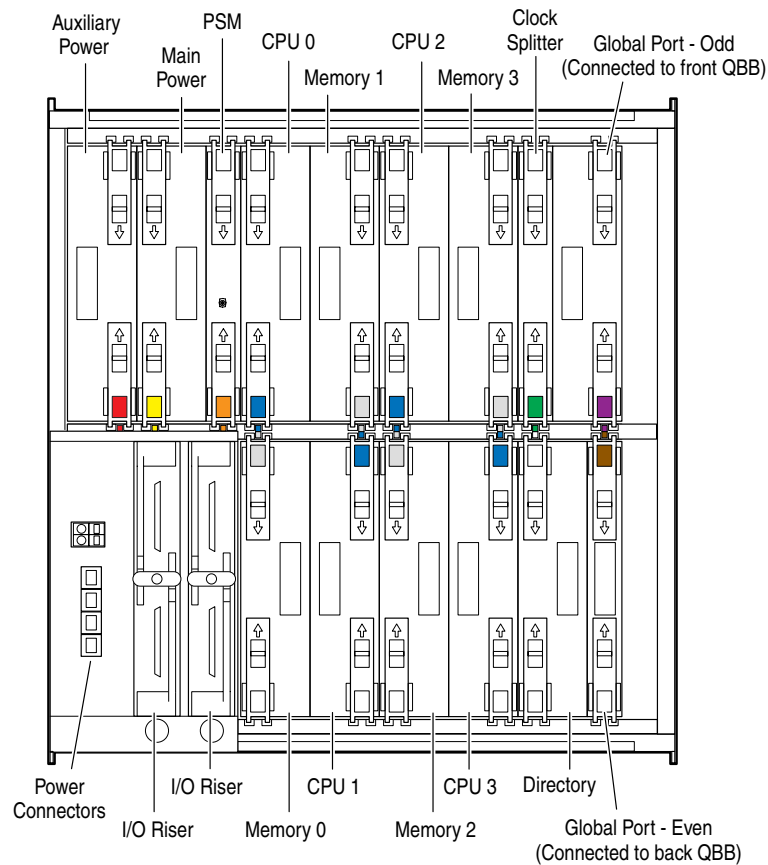




### System Box Configuration Rules

- A system box has two QBBs.
- A QBB supports up to four CPU modules.
- A QBB supports up to four memory modules.
- A QBB has up to two I/O riser modules; each I/O riser module connects to one PCI box.
- A system box supports up to four PCI boxes.

**Figure 3–6 System Box QBB (Cabinet Rear)**

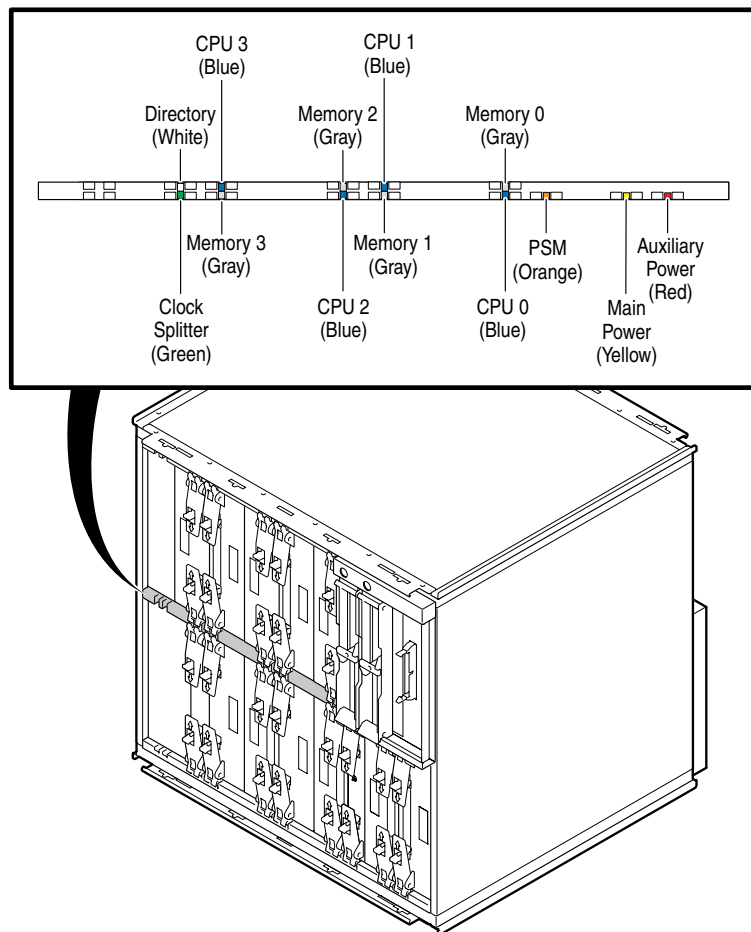


PK2228

## 3.5 QBB Color Code

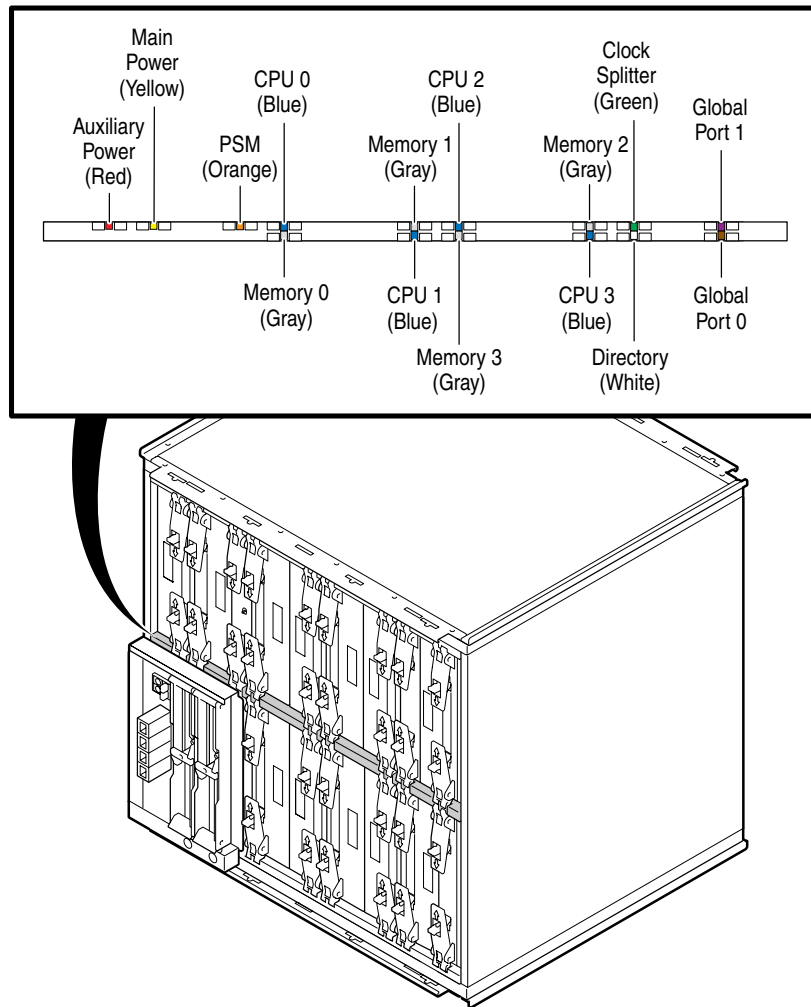
Figure 3-7 and Figure 3-8 show the center bar color code for module placement in the QBB. Note that CPU and memory slots are color-coded to ensure the correct placement of each module.

Figure 3-7 QBB Center Bar Color Code (Cabinet Front)



PK0628

**Figure 3-8 QBB Center Bar Color Code (Cabinet Rear)**

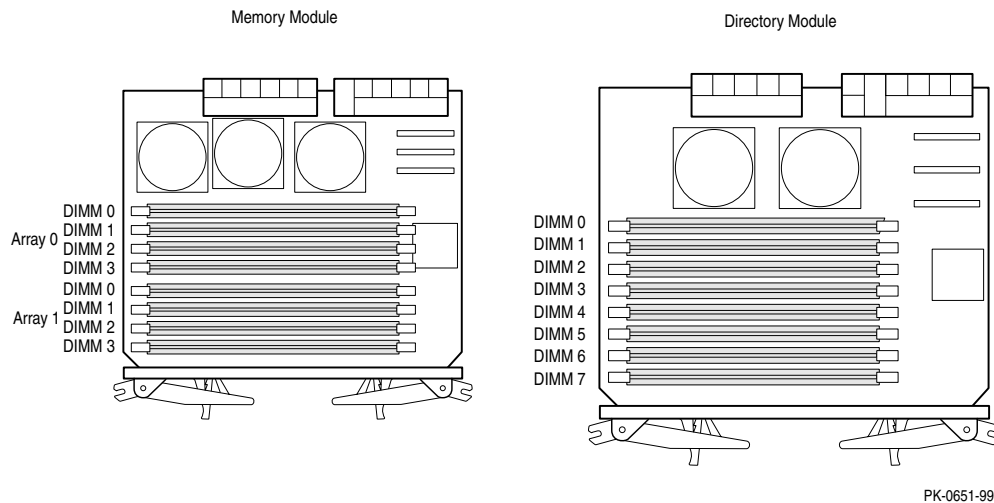


PK2229

## 3.6 Memory Configurations

A memory module has eight DIMM slots. Two arrays (Array 0 and Array 1), consisting of four DIMMs, can be installed on each module. A directory DIMM is required for each array in systems having more than four processors. Directory DIMMs are installed on the directory module.

Figure 3-9 Memory Module and Directory Module



### Memory Configuration Guidelines

- On a memory module, DIMMs are divided into two groups of four called arrays.
- A memory module must be populated on an array-by-array basis; that is, groups of four DIMMs must be installed.
- DIMMs in an array must be the same size and type.
- DIMM sizes include 256 Mbyte, 512 Mbyte, and 1 Gbyte.

There are two types of DIMMs: single density (SD) and double density (DD).

- Density does not affect interleaving.
- One directory module DIMM is required for each memory array in systems with more than one QBB.
- The type and number of DIMMs installed in the directory module is based on the number of memory arrays populated and the physical size of the memory arrays. See the table below.
- A larger DIMM type may always be used in place of the minimum required DIMM type.

Directory DIMM Type	Memory Array Size	Part Number
0	256 MB	54-25019-AA
1	1 GB	54-25023-AA
2	2 GB	54-25023-BA
3	2 GB	54-25023-CA
4	4 GB	54-25023-DA

## 3.7 Memory Interleaving Guidelines

**Table 3-1 Interleaving Memory Modules**

Memory Interleaving	Guidelines
4-way	The default interleave. One memory module with one array populated (or most mixes not discussed below).
8-way	One memory module with two arrays populated.  Preferred method: Two memory modules with one array populated on each module.
16-way	Two memory modules, each with two arrays populated.  Preferred method. Four memory modules with one array populated on each module.
32-way	Four memory modules with two arrays populated on each module.

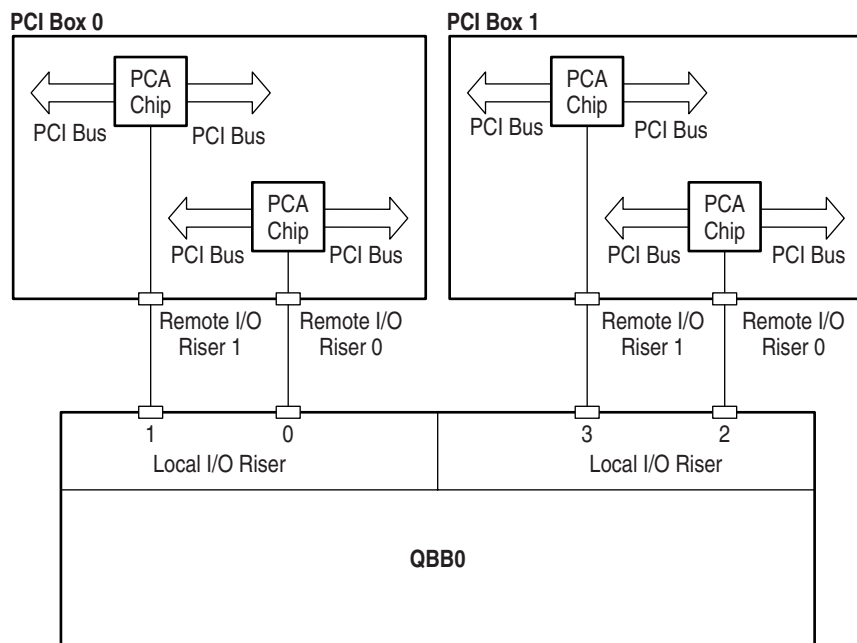
### **Memory Interleaving Guidelines**

- The larger the interleaving factor, the better the system performance.
- Avoid mixing memory sizes; this limits interleaving capability and potential bandwidth.

## 3.8 PCI Boxes

A QBB supports up to two PCI boxes. A cable connects the QBB “local” I/O riser to the “remote” I/O riser in the PCI box. There are two I/O ports on a local I/O riser. Each I/O port is used to connect to one remote I/O riser. Figure 3-10 shows QBB0 connected to PCI box 0 and PCI box 1.

Figure 3-10 Sample I/O Subsystem



PK-0652-99



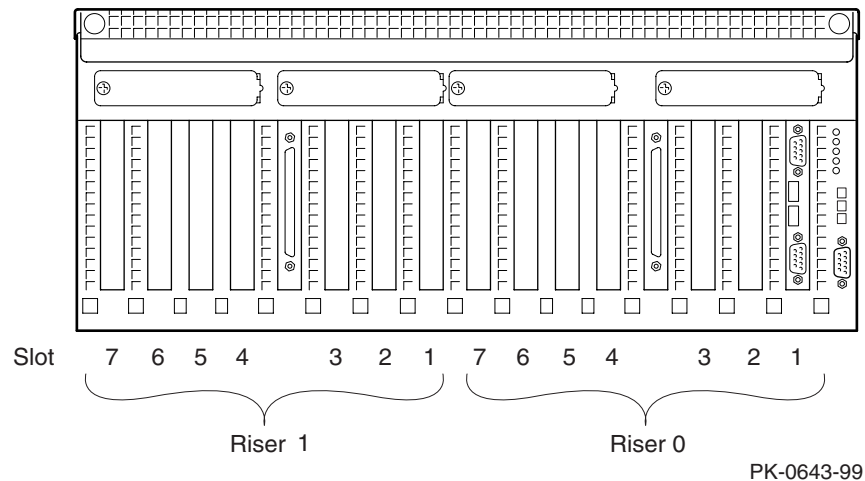
The I/O subsystem consists of the local I/O interface (QBB) and the remote I/O interface (PCI box) connected by I/O cables.

A system can have up to 16 PCI boxes. To identify PCI boxes in a system, a node ID is set using the node ID switch located on the rear panel of each PCI box (see Figure 3–11).

### 3.9 PCI Box Slot Configuration

Each QBB can have two I/O risers supporting up to two PCI boxes. A cable connects a local I/O riser (in the QBB) to a remote I/O riser (in the PCI box). Each PCI box can have up to two remote I/O risers in place. Cable connectors for the two remote I/O risers are shown as Riser 0 and Riser 1 in Figure 3-11. PCI slots and logical hoses are listed in Table 3-2.

Figure 3-11 PCI Slot Locations



## PCI Slot Configuration Guidelines

- I/O riser 0 must be installed.
- The standard I/O module is always installed in riser 0-slot 1.
- Install high-powered modules in slots with one inch module pitch (all slots except riser 0-slot 5, riser 0-slot 6, riser 1-slot 5, and riser 1-slot 6).
- Install high-performance adapters across multiple bus/hose segments to get maximum performance.
- VGA graphics options must be installed in riser 0-slot 2 or riser 0-slot 3.

---

**CAUTION:** *Installing a full-length module next to the standard I/O module requires extra care due to cabling on the standard I/O module.*

---

## Logical Hoses

You can have a maximum of four logical hoses per PCI box. Logical hose numbers are assigned by the firmware. Logical hoses are numbered from 0 to 63.

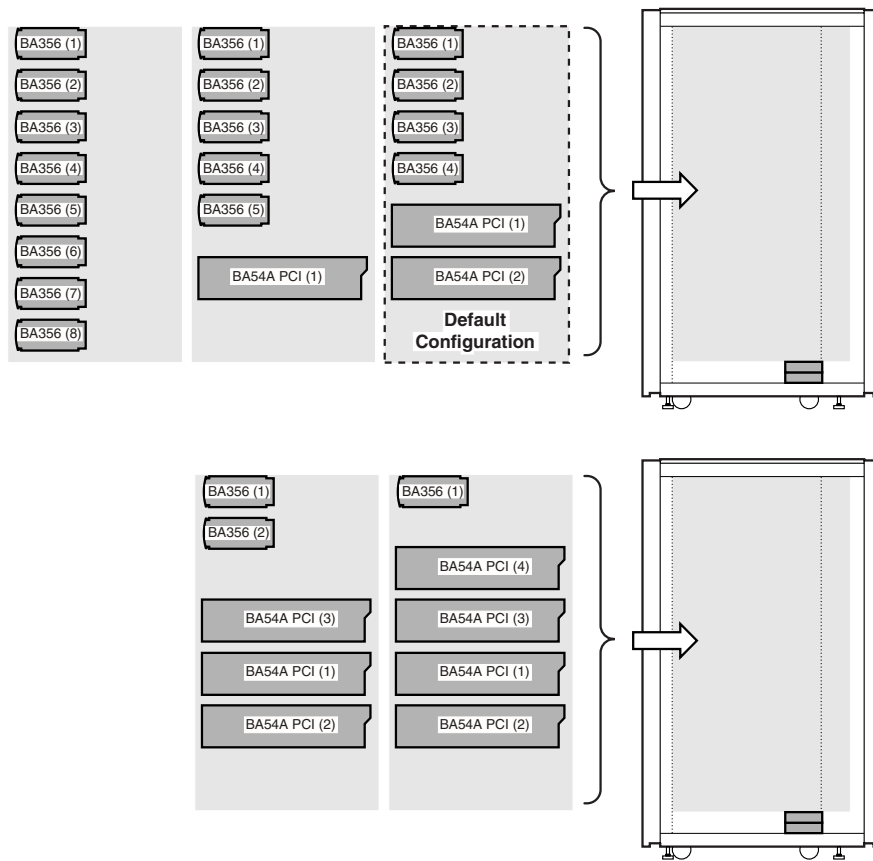
**Table 3-2 PCI Slots and Logical Hoses**

Remote I/O Riser 0		Remote I/O Riser 1	
Logical Hose 0	Logical Hose 1	Logical Hose 2	Logical Hose 3
Riser 0-Slot 1	Riser 0-Slot 4	Riser 1-Slot 1	Riser 1-Slot 4
Riser 0-Slot 2	Riser 0-Slot 5	Riser 1-Slot 2	Riser 1-Slot 5
Riser 0-Slot 3	Riser 0-Slot 6	Riser 1-Slot 3	Riser 1-Slot 6
---	Riser 0-Slot 7	---	Riser 1-Slot 7

## 3.10 Expander Cabinet

Additional PCI boxes and storage devices are housed in an expander cabinet. The same cabinet is used to expand GS160/320 systems and GS80 systems. Figure 3-12 shows five different PCI and BA356 storage configurations.

Figure 3-12 BA356 Storage Device Configurations



PK-0646-99

# Chapter 4

## GS80 Rack System Overview

In the rack system, the BA52A system drawer has a QBB containing a backplane, CPU modules, memory modules, power modules, and I/O riser modules.

This chapter provides an overview of the BA52A drawer in these sections:

- Rack System Characteristics
- System Drawer Architecture
- System Drawer Modules

## 4.1 Rack System Characteristics

Table 4-1 lists system drawer characteristics. Table 4-2 lists power and environmental specifications for the rack system.

Figure 4-1 System Drawer

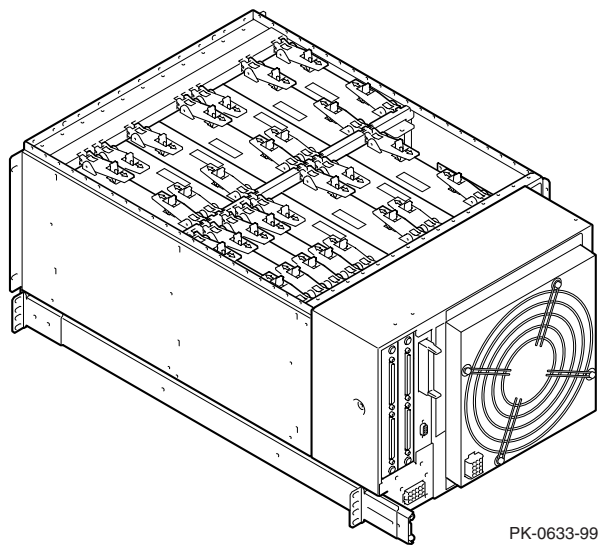


Table 4-1 System Drawer Characteristics

Characteristic	Specification
Size	40 cm H x 45 cm W x 65 cm D (15 in. x 18 in. x 25 in.)
Weight	45 kg (100 lb) maximum
Maximum heat dissipation	1500 w (4910 Btu/hr)

**Table 4-2 Rack System Characteristics**

---

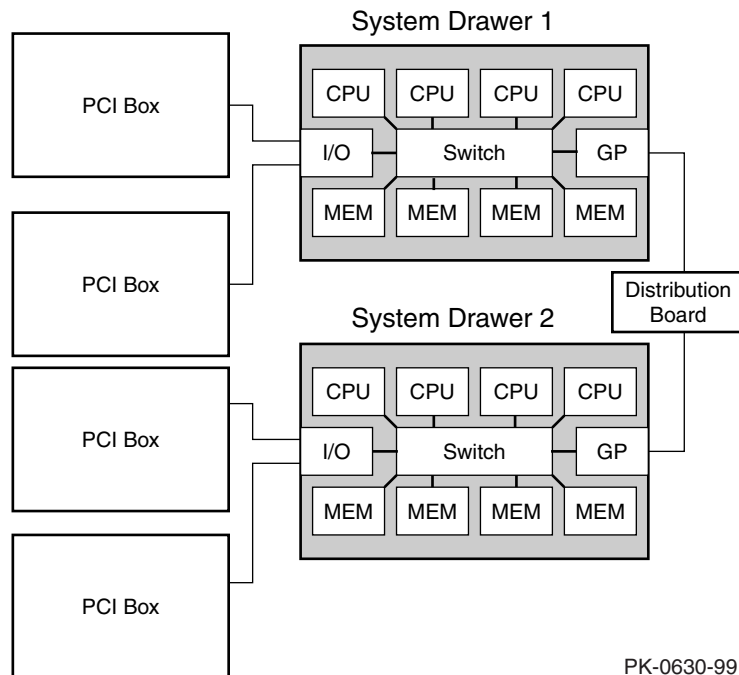
<b>Electrical</b>	
Voltage	120 VAC (U.S.) 220–240 VAC (Europe) 200–240 VAC (Japan)
Phase	Single
Frequency	50–60 Hz
Maximum input current/circuit	16 A (U.S.) 12 A (Europe) 13 A (Japan)
Maximum power consumption	2.4 – 2.8 KVA (U.S.) 5.2 – 5.7 KVA (Europe) 4.8 – 5.7 KVA (Japan)
<b>Environmental</b>	
Temperature	Operating: 5° to 35°C (41 to 95°F) Not operating: -40° to 66°C (-40° to 150°F)
Humidity	Operating: 10% to 90% Not operating: 10% to 95%
Altitude	Operating: 0 to 3 km (0 to 10,000 ft) Not operating: 0 to 12.2 km (0 to 40,000 ft)

---

## 4.2 System Drawer Architecture

The system drawer houses a QBB consisting of a backplane that supports four CPU modules, four memory modules, two power modules and two I/O riser modules. These modules are identical to those used in the box systems. The global port is part of the backplane. In a two-drawer system, the drawers are linked by a distribution board.

Figure 4-2 Two-Drawer Block Diagram





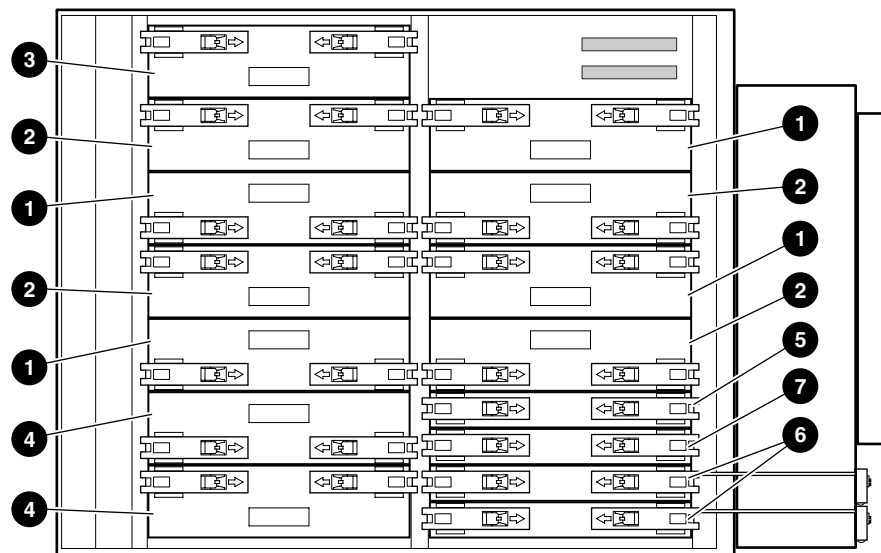
The switch that interconnects the CPU modules, memory modules, and I/O riser modules is built into the system drawer backplane.

In a two-drawer system, the system drawers are linked together through the global ports and the distribution board. A directory module is required in each system drawer in a two-drawer system.

## 4.3 System Drawer Modules

The modules plug into the system drawer backplane. Figure 4–3 shows a fully populated backplane. Figure 4–4 shows the backplane with no modules.

Figure 4–3 System Drawer Modules

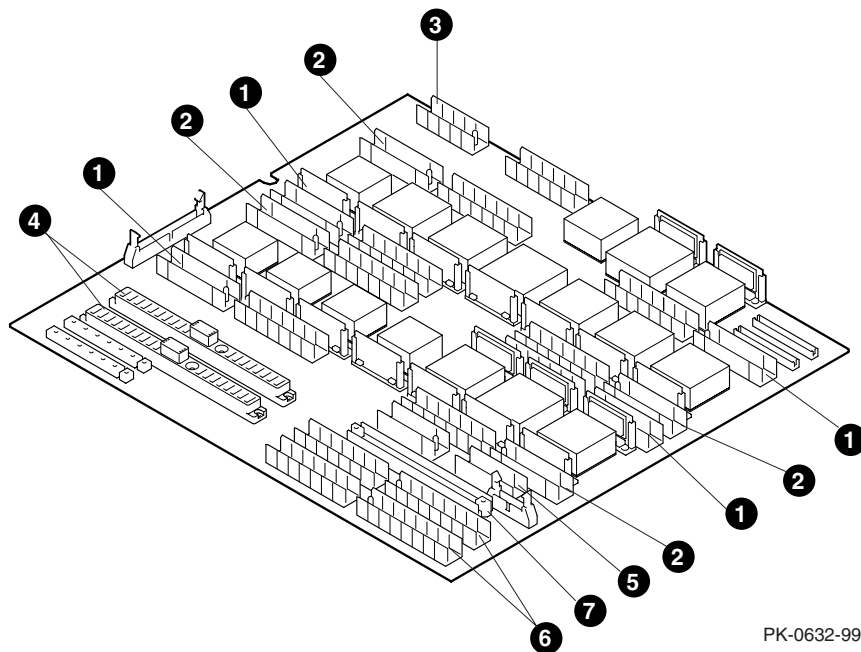


PK0619

The CPU, memory, power, and I/O riser modules plug into the backplane located at the bottom of the system drawer. Callouts in Figure 4–3 point to the location of the following system drawer modules:

- ❶ CPU module
- ❷ Memory module
- ❸ Directory module
- ❹ Power modules
- ❺ Power system manager (PSM) module
- ❻ I/O riser modules
- ❼ Clock splitter module

**Figure 4–4 System Drawer Backplane**





# Chapter 5

## GS80 Rack System Configuration Rules

This chapter provides configuration rules for the following:

- Rack
- Rack Power System

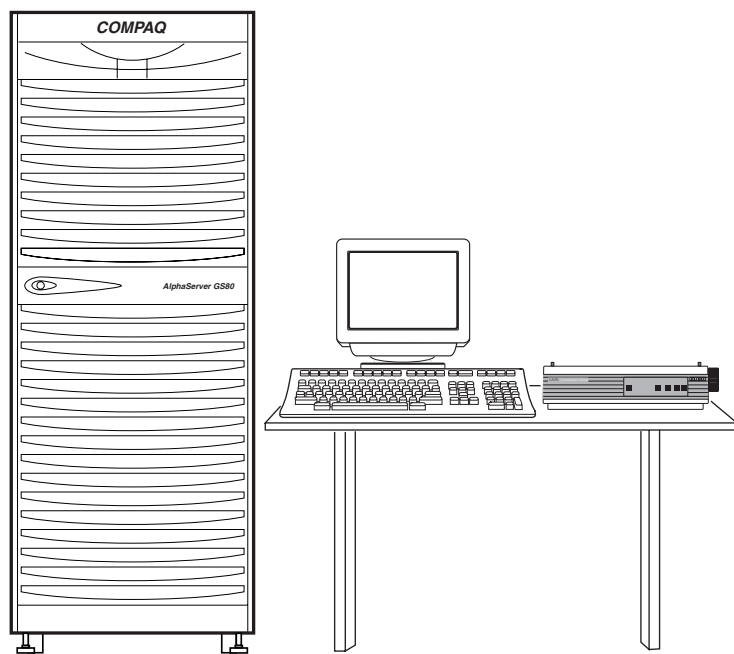
## 5.1 Rack

---

**A rack houses a maximum of two system drawers.**

---

**Figure 5-1 Rack**



PK-0635-99

### **About the Rack Cabinet**

The cabinet contains the following components:

- One or two system drawers
- Control panel (see Section 2.6 for details on the control panel)
- AC input box
- Power subrack with power supplies
- Vertical mounting rails
- Stabilizer
- Wrist strap for static discharge protection

### **Rack Variants**

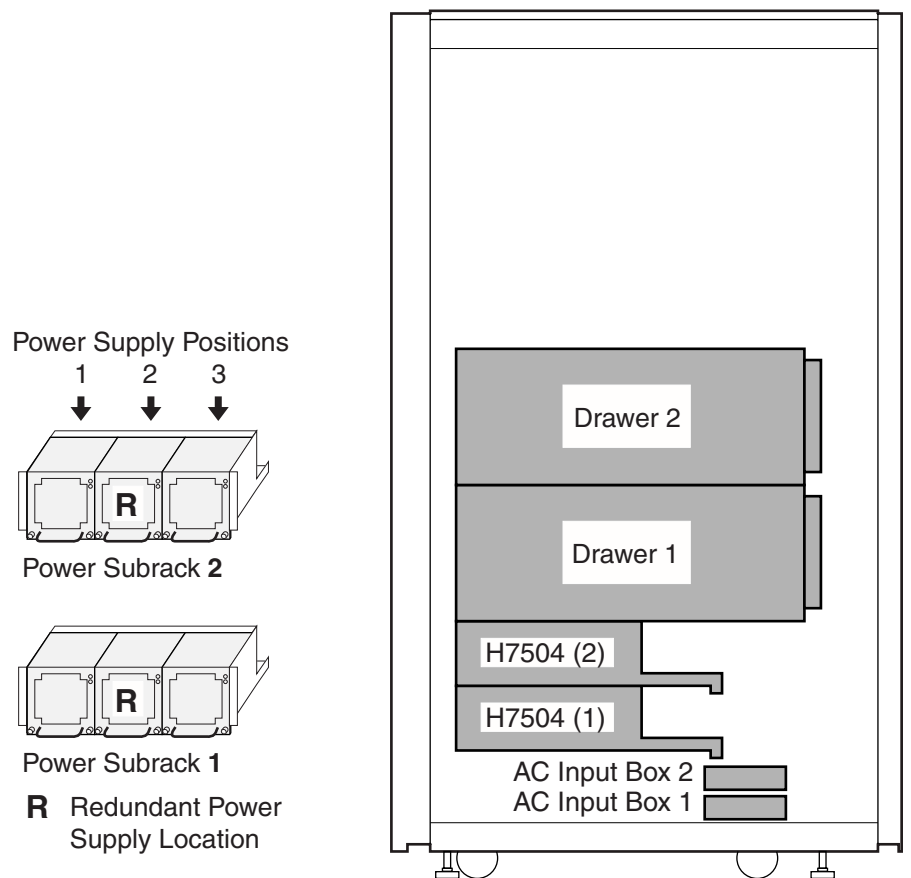
The H9A20 RETMA cabinet has three variants:

- H9A20-CA, North American (120V)
- H9A20-CB, European (220–240V)
- H9A20-CC, North American and Japanese (200–240V)

## 5.2 Rack Power System

Figure 5–2 shows a two-drawer rack power system: two AC input boxes and two H7504 power subracks at the bottom of the cabinet. Each subrack holds three power supplies. The system drawer power cables connect to the power subrack.

Figure 5–2 Two-Drawer Rack Power System



PK-0638-99



### **About the Power System**

- Each system drawer requires one power subrack.
- Each system drawer requires two power supplies.
- Each power subrack holds up to three power supplies. The third power supply is used for redundancy.



# Chapter 6

## Booting and Installing an Operating System

This chapter provides basic operating instructions, including powering up the system and booting the operating system.

Sections in this chapter are:

- Powering Up the System
- Setting Boot Options
- Booting *Tru64 UNIX*
- Installing *Tru64 UNIX*
- Booting *OpenVMS*
- Installing *OpenVMS*

## 6.1 Powering Up the System

To power up the system, set the keyswitch to On, or power up the system remotely. The SCM power-up display is shown at the system management console and the control panel, followed by the SRM power-up display.

### 6.1.1 SCM Power-Up Display

#### Example 6-1 SCM Power-Up Display

```
SCM_E0> power on
Powering on PCI Box 0
Powering on PCI Box 1
QBB-0 Powering ON

~I~ Testing OCP Switch passed

Power ON Phase INIT
QBB-1 Powering ON
QBB-2 Powering ON
QBB-3 Powering ON

SCM_E0>
Testing SIO Shared RAM(please wait)
Initializing shared ram
Shared RAM Initialized

Powering ON H-Switch
SCM_E0>
I~ HSW4/HPM40 SysEvent: HS_INIT_CD1                      Reg0:000F Reg1:AB81

Phase 0
I~ Enable HS Links: 0f

~I~ QbbConf(gp/io/c/m)=fbbfffff Assign=ff SQbb0=00 PQbb=00 SoftQbbId=fedcba98
~I~ SysConfig: 37 13 07 19 07 12 c7 13 37 13 f7 11 f7 13 37 13

SCM_E0>
I~ HSW4/HPM40 SysEvent: LINK0_ON                      Reg0:000F Reg1:AB81
SCM_E0>
I~ HSW4/HPM40 SysEvent: LINK1_ON                      Reg0:010F Reg1:AB81
SCM_E0>
I~ HSW4/HPM40 SysEvent: LINK2_ON                      Reg0:030F Reg1:AB81
SCM_E0>
I~ HSW4/HPM40 SysEvent: LINK3_ON                      Reg0:070F Reg1:AB81
SCM_E0>
.....
SCM_E0> .....
```

❶

❷

❸

- ❶ The user issues a **power on** command.
- ❷ Messages denoted by ~I~ are informational and do not indicate a serious event. Other types of messages include:
  - \*\*\* – Diagnostic format indicating an error has occurred.
  - ### – Diagnostic format indicating a warning.
  - ~E~ – An error has occurred; power-up continues, but the affected resource is dropped.
  - ~W~ – An error has occurred; power-up continues, and the affected resource is questionable.
- ❸ During the Phase INIT (initialization phase) SROM code is loaded into each CPU in the system and communication between the power system manager (PSM) and the CPU is established. Phase 0 follows with local QBB testing.

*Continued on next page*

## Example 6-1 SCM Power-Up Display (Continued)

```
QBB2 Step(s)-0 1 2 3 4 5 Tested
QBB3 Step(s)-0 1 2 3 4 5 Tested
QBB0 Step(s)-0 1 2 3 4 5 Tested
QBB1 Step(s)-0 1 2 3 4 5 Tested
```

Phase 1

```
QBB0 IO_MAP0: 000000C101311133
QBB1 IO_MAP1: 0000000000000003
QBB2 IO_MAP2: 0000000000000003
QBB3 IO_MAP3: 000000C001311133
```

```
~I~ QbbConf(gp/io/c/m)=fbbfffff Assign=ff SQbb0=00 PQbb=00 SoftQbbId=fedcba98
```

```
~I~ SysConfig: 37 13 07 19 07 12 c7 13 37 13 f7 11 f7 13 37 13
```

SCM\_E0>

```
QBB1 now Testing Step-6
QBB1 now Testing Step-7
QBB1 now Testing Step-8
QBB1 now Testing Step-9
QBB1 now Testing Step-A
QBB2 now Testing Step-6
QBB2 now Testing Step-7
QBB3 now Testing Step-6
QBB3 now Testing Step-7
QBB3 now Testing Step-8
QBB3 now Testing Step-9
QBB3 now Testing Step-A.
QBB0 now Testing Step-6
QBB1 Step(s)-A B Tested
QBB2 Step(s)-7 8 9 A B Tested
QBB3 Step(s)-A B Tested.
QBB0 now Testing Step-7.
QBB0 now Testing Step-8..
QBB0 now Testing Step-9...
QBB0 now Testing Step-A..
QBB0 now Testing Step-7.
QBB0 now Testing Step-8...
QBB0 now Testing Step-9..
QBB0 now Testing Step-A..
QBB0 now Testing Step-7.
QBB0 now Testing Step-8...
QBB0 now Testing Step-9...
QBB0 now Testing Step-A..
QBB0 now Testing Step-B..
```

Phase 2

```
QBB0 IO_MAP0: 000000C101311133
QBB1 IO_MAP1: 0000000000000003
QBB2 IO_MAP2: 0000000000000003
QBB3 IO_MAP3: 000000C001311133
```

```
~I~ QbbConf(gp/io/c/m)=fbbfffff Assign=ff SQbb0=00 PQbb=00 SoftQbbId=fedcba98
```

```
~I~ SysConfig: 37 13 07 19 07 12 c7 13 37 13 f7 11 f7 13 37 13
```

SCM\_E0>

```
QBB1 now Testing Step-C
QBB2 now Testing Step-C
QBB3 now Testing Step-C
QBB0 Step(s)-B C Tested...
```

## Example 6-1 SCM Power-Up Display (Continued)

```
Phase 3 ⑥
~I~ QbbConf(gp/io/c/m)=fbbffffff Assign=ff SQbb0=00 PQbb=00 SoftQbbId=fedcba98
~I~ SysConfig: 37 13 07 19 07 12 c7 13 37 13 f7 11 f7 13 37 13
SCM_E0>
QBB0 now Testing Step-D
QBB1 now Testing Step-D
QBB2 now Testing Step-D
QBB3 now Testing Step-D.....
QBB0 IO_MAP0: 000000C101311133
QBB1 IO_MAP1: 0000000000000003
QBB2 IO_MAP2: 0000000000000003
QBB3 IO_MAP3: 000000C001311133

Phase 4 ⑦
~I~ QbbConf(gp/io/c/m)=fbbffffff Assign=ff SQbb0=00 PQbb=00 SoftQbbId=fedcba98

QBB0 unloading console across port0 from PCI Box-1
Console COM1 from master PCI Box-0
~I~ SysConfig: 37 13 07 19 07 12 c7 13 37 13 f7 11 f7 13 37 13
Retrieving FRU information for Shared RAM...
SCM_E0>
QBB0 now Testing Step-E..
Power On Complete

Returning to system COM1 port ⑧
```

- ④ Phase 1. The primary CPU, selected by the SCM in phase 0, tests each QBB in the system.
- ⑤ Phase 2. Secondary CPUs are tested to ensure cache coherency.
- ⑥ Phase 3. The tests ensure that each CPU can access each memory array in the system.
- ⑦ Phase 4. The primary CPU unloads the PAL/console code from the flash ROM on the standard I/O module into memory.
- ⑧ Control of the remainder of power-up is passed to the console firmware. See Section 6.1.2.

## 6.1.2 SRM Power-Up Display

**Following the initial SCM power-up and the five test phases, the SRM console takes control of the remaining portion of system power-up.**

### Example 6-2 SRM Power-Up Display

System Primary QBB0 : 0  
System Primary CPU : 0 on QBB0

Par	hrd/sft	CPU	Mem	IOR3	IOR2	IOR1	IOR0	GP	QBB	Dir	PS	Temp
QBB#	3210	3210		(pci_box.rio)				Mod	BP	Mod	321	(:C)
(-)	0/30	---P	PP	---	---	Px.x	P1.0	P	P	P	PPP	26.5
(-)	1/31	-P-P	PP	---	---	---	---	P	P	P	PPP	29.0
(-)	2/32	--P-	--P-	---	---	---	---	P	P	P	PP-	25.0
(-)	3/33	---P	-P--	---	---	Px.x	P0.0	P	P	P	PP-	27.0

❶

HSwitch	Type	Cables	7	6	5	4	3	2	1	0	Temp(:C)
HPM40	4-port		-	-	-	-	E	E	E	E	29.0

PCI	Rise1-1	Rise1-0	Rise0-1	Rise0-0	RIO	PS	Temp							
Cab	7	6	5	4	3	2	1	0	21	(:C)				
10	-	-	-	-	-	-	S	-	*	P-	27.5			
11	-	-	-	-	-	L	-	L	M	S	-	*	PP	27.0

OpenVMS PALcode V1.75-1, Tru64 UNIX PALcode V1.68-1

❷

```
system = QBB 0 1 2 3          + HS
QBB 0 = CPU 0          + Mem 0 1          + Dir + IOP + PCA 0          + GP (Hard QBB 0)
QBB 1 = CPU 0          2          + Mem 0 1          + Dir + IOP + PCA          + GP (Hard QBB 1)
QBB 2 = CPU          1          + Mem          1          + Dir + IOP + PCA          + GP (Hard QBB 2)
QBB 3 = CPU 0          + Mem          2          + Dir + IOP + PCA 0          + GP (Hard QBB 3)
```

❸

```
shared RAM version is 1.3
hose 24 has a standard I/O module
starting console on CPU 0
initialized idle PCB
initializing semaphores
initializing heap
initial heap 2c0c0
memory low limit = 1f2000
heap = 2c0c0, 1ffc0

initializing driver structures
initializing idle process PID
initializing file system
initializing timer data structures
lowering IPL
```



- ❶ A snapshot of the system environment is displayed. See Section 8.7.3 for more information.
- ❷ PALcode is loaded and started.
- ❸ The size of the system is determined and mapped. This system has four QBBs and five CPUs.

*Continued on next page*

## Example 6-2 SRM Power-Up Display (Continued)

```
CPU 0 speed is 731 MHz
create dead_eater
create poll
create timer
create powerup
access NVRAM

QBB 0 memory, 1 GB
QBB 1 memory, 1 GB
QBB 2 memory, 512 MB
QBB 3 memory, 512 MB
total memory, 3 GB
copying PALcode to 103ffe8000
copying PALcode to 201ffe8000
copying PALcode to 301ffe8000

probe I/O subsystem
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
bus 1, slot 0 -- dva-Floppy
bus 0, slot 1 -- pka-QLogic ISP10x0
bus 0, slot 2 -- vga-ELSA GLoria Synergy
bus 0, slot 3 -- pkb-QLogic ISP10x0
bus 0, slot 15 -- dqa-Acer Labs M1543C IDE
bus 0, slot 15 -- dqb-Acer Labs M1543C IDE
probing hose 1, PCI
bus 0, slot 5 -- ewa-DE500-BA Network Controller
probing hose 24, PCI
probing PCI-to-ISA bridge, bus 1
bus 1, slot 0 -- dvb-Floppy
bus 0, slot 1 -- pkc-QLogic ISP10x0
bus 0, slot 15 -- dqc-Acer Labs M1543C IDE
bus 0, slot 15 -- dqd-Acer Labs M1543C IDE
probing hose 25, PCI
starting drivers
*** keyboard not plugged in...
entering idle loop
initializing keyboard
*** keyboard not plugged in...

starting console on CPU 4
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 4 speed is 731 MHz
create powerup
starting console on CPU 6
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 6 speed is 731 MHz
create powerup
starting console on CPU 9
initialized idle PCB
initializing idle process PID
lowering IPL
```

*Continued on next page*

## Example 6-2 SRM Power-Up Display (Continued)

```
CPU 9 speed is 731 MHz
create powerup
starting console on CPU 12
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 12 speed is 731 MHz
create powerup
initializing pka pkb pkc ewa dqa dqb dqc dqd
initializing GCT/FRU at 1f2000
AlphaServer Console X5.7-6290, built on Feb  4 2000 at 01:41:06
P00>>>
```

7

- ④ Distributed memory is sized and mapped.
- ⑤ The I/O subsystem is mapped.
- ⑥ Each secondary CPU starts the console, is initialized and ready to join the multiprocessor environment.
- ⑦ The SRM console prompt is displayed.

## 6.2 Setting Boot Options

---

You can set a default boot device, boot flags, and network boot protocols for *Tru64 UNIX* or *OpenVMS* using the SRM **set** command with environment variables. Once these environment variables are set, the boot command defaults to the stored values. You can override the stored values for the current boot session by entering parameters on the boot command line.

---

The SRM boot-related environment variables are listed below and described in the following sections:

<b>bootdef_dev</b>	Defines a default boot device.
<b>boot_file</b>	Specifies a default file name to be used for booting when no file name is specified by the <b>boot</b> command.
<b>boot_osflags</b>	Defines parameters to enable specific functions during the boot process.
<b>ei*0_inet_init</b> or <b>ew*0_inet_init</b>	Determines whether the interface's internal Internet database is initialized from nvram or from a network server (through the bootp protocol). Set this environment variable if you are booting <i>Tru64 UNIX</i> from a RIS server.
<b>ei*0_protocols</b> or <b>ew*0_protocols</b>	Defines a default network boot protocol (bootp or mop).

### 6.2.1 Bootdef\_dev

---

The **bootdef\_dev** environment variable specifies one or more devices from which to boot the operating system. When more than one device is specified, the system searches in the order listed and boots from the first device.

---

Enter the **show bootdef\_dev** command to display the current default boot device. Enter the **show device** command for a list of all devices in the system.

The syntax is:

**set bootdef\_dev** *boot\_device*

*boot\_device*    The name of the device on which the system software has been loaded. To specify more than one device, separate the names with commas.

### Example

In this example, two boot devices are specified. The system will try booting from dkb0 and, if unsuccessful, will boot from dka0.

```
P00>>> set bootdef_dev dkb0, dka0
```

---

**NOTE:** *When you set the **bootdef\_dev** environment variable, it is recommended that you set the operating system boot parameters as well, using the **set boot\_osflags** command.*

---

## 6.2.2 Boot\_file

---

**The boot\_file environment variable specifies the default file name to be used for booting when no file name is specified by the boot command.**

---

The syntax is:

**set boot\_file** *filename*

### Example

In this example, a boot file is specified for booting *OpenVMS* from the InfoServer. APB\_0712 is the file name of the APB program used for the initial system load (ISL) boot program.

```
P00>>> set boot_file apb_0712
```

### 6.2.3 Boot\_osflags

---

**The `boot_osflags` environment variable sets the default boot flags and, for *OpenVMS*, a root number.**

---

Boot flags contain information used by the operating system to determine some aspects of a system bootstrap. Under normal circumstances, you can use the default boot flag settings.

To change the boot flags for the current boot only, use the *flags\_value* argument with the **boot** command.

The syntax is:

**set boot\_osflags** *flags\_value*

The *flags\_value* argument is specific to the operating system.

#### **Tru64 UNIX Systems**

*Tru64 UNIX* systems take a single ASCII character as the *flags\_value* argument.

- a** Load operating system software from the specified boot device (autoboot). Boot to multiuser mode.
- i** Prompt for the name of a file to load and other options (boot interactively). Boot to single-user mode.
- s** Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt.
- D** Full dump; implies “s” as well. By default, if *Tru64 UNIX* crashes, it completes a partial memory dump. Specifying “**D**” forces a full dump at system crash.

## OpenVMS Systems

*OpenVMS* systems require an ordered pair as the *flags\_value* argument: *root\_number* and *boot\_flags*.

*root\_number*      Directory number of the system disk on which *OpenVMS* files are located. For example:

<i>root_number</i>	Root Directory
--------------------	----------------

0 (default)	[SYS0.SYSEXE]
-------------	---------------

1	[SYS1.SYSEXE]
---	---------------

2	[SYS2.SYSEXE]
---	---------------

3	[SYS3.SYSEXE]
---	---------------

*boot\_flags*      The hexadecimal value of the bit number or numbers set. To specify multiple boot flags, add the flag values (logical OR). For example, the flag value 10080 executes both the 80 and 10000 flag settings. See Table 6–1.

**Table 6–1   OpenVMS Boot Flag Settings**

Flags_Value	Bit Number	Meaning
1	0	Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT).
2	1	Map XDELTA to a running system.
4	2	Stop at initial system breakpoint.
8	3	Perform diagnostic bootstrap.
10	4	Stop at the bootstrap breakpoints.
20	5	Omit header from secondary bootstrap image.
80	7	Prompt for the name of the secondary bootstrap file.
100	8	Halt before secondary bootstrap.
10000	16	Display debug messages during booting.
20000	17	Display user messages during booting.

*Continued on next page*

### Example

In the following *Tru64 UNIX* example, the boot flags are set to autoboot the system to multiuser mode when you enter the **boot** command.

```
P00>>> set boot_osflags a
```

In the following *OpenVMS* example, *root\_number* is set to 2 and *boot\_flags* is set to 1. With this setting, the system will boot from root directory SYS2.SYSEXEC to the SYSBOOT prompt when you enter the **boot** command.

```
P00>>> set boot_osflags 2,1
```

In the following *OpenVMS* example, *root\_number* is set to 0 and *boot\_flags* is set to 80. With this setting, you are prompted for the name of the secondary bootstrap file when you enter the **boot** command.

```
P00>>> set boot_osflags 0,80
```

### 6.2.4 ei\*0\_inet\_init or ew\*0\_inet\_init

---

**The ei\*0\_inet\_init or ew\*0\_inet\_init environment variable determines whether the interface's internal Internet database is initialized from nvram or from a network server (through the bootp protocol).**

---

Legal values are **nvram** and **bootp**. The default value is **bootp**. Set this environment variable if you are booting *Tru64 UNIX* from a RIS server.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters “ei” or “ew,” for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (\*) with the adapter ID letter when entering the command.

The syntax is:

```
set ei*0_inet_init value or  
set ew*0_inet_init value
```

#### Example

```
P00>>> set eia0_inet_init bootp
```



## 6.2.5 ei\*0\_protocols or ew\*0\_protocols

---

The **ei\*0\_protocols** or **ew\*0\_protocols** environment variable sets network protocols for booting and other functions.

---

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters “ei” or “ew,” for example, eia0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (\*) with the adapter ID letter when entering the command.

The syntax is:

**set ei\*0\_protocols** *protocol\_value* or  
**set ew\*0\_protocols** *protocol\_value*

The options for *protocol\_value* are:

- |                      |   |
|----------------------|---|
| <b>mop</b> (default) | Sets the network protocol to mop (Maintenance Operations Protocol), the setting typically used with the <i>OpenVMS</i> operating system.                              |
| <b>bootp</b>         | Sets the network protocol to bootp, the setting typically used with the <i>Tru64 UNIX</i> operating system.   |
| <b>bootp,mop</b>     | When both are listed, the system attempts to use the mop protocol first, regardless of which is listed first. If not successful, it then attempts the bootp protocol. |

### Example

```
P00>>> show device
.
.
.
eia0.0.0.1001.0      EIA0      08-00-2B-3E-BC-B5
eib0.0.0.12.0       EIB0      00-00-C0-33-E0-0D
eic0.0.0.13.0       EIC0      08-00-2B-E6-4B-F3
.
.
.
P00>>> set eia0_protocols bootp
P00>>> show eia0_protocols
eia0_protocols      bootp
```

## 6.3 Booting Tru64 UNIX

---

**Tru64 UNIX can be booted from a DVD or CD-ROM on a local drive, from a local SCSI disk, or from a server.**

---

### Example 6-3 Booting Tru64 UNIX from a Local SCSI Disk

```
P00>>> sho dev ❶
dka0.0.0.1.0          DKA0          RZ2ED-LS  0306
dka100.1.0.1.1        DKA100        RZ2ED-LS  0306
dka200.2.0.1.1        DKA200        RZ2DD-LS  0306
dka300.3.0.1.1        DKA300        RZ2DD-LS  0306
dkc0.0.0.1.0          DKC0          RZ2DD-LS  0306
dkc100.1.0.1.0        DKC100        RZ2DD-LS  0306
dkc200.2.0.1.0        DKC200        RZ2DD-LS  0306
dkc300.3.0.1.0        DKC300        RZ2DD-LS  0306
dqa0.0.0.15.0         DQA0          TOSHIBA CD-ROM XM-6202B 1110
dva0.0.0.1000.0       DVA0

P00>>> boot ❷
(block dka0.0.0.1.0 -flags a) ❸
block 0 of dka0.0.0.1.0 is a valid boot block
reading 13 blocks from dka0.0.0.1.0
bootstrap code read in
base = 2e6000, image_start = 0, image_bytes = 1a00
initializing HWRPB at 2000
initializing page table at 17fbda000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Wed Sep 8 20:59:58 EDT 1999

Loading vmunix ...
Loading at 0xfffffc0000430000

Sizes:
text = 4800560
data = 903872
bss = 1412480
Starting at 0xfffffc00007790d0

Loading vmunix symbol table ... [1330800 bytes]
Alpha boot: available memory from 0x93de000 to 0x3180000000
Digital Tru64 UNIX V4.0G-1 (Rev. 1398); Thu Oct 21 13:34:00 EDT 1999
physical memory = 24572.00 megabytes.
available memory = 24140.65 megabytes.
using 94344 buffers containing 737.06 megabytes of memory
Master cpu at slot 0.
```

### Example 6–3 Booting Tru64 UNIX from a Local SCSI Disk (Continued)

```
Firmware revision: 5.6-6930
PALcode: Digital Tru64 UNIX version 1.60-1
Compaq AlphaServer GS320 6/731
.
.
.
Digital Tru64 UNIX Version V4.0
login:
```

Example 6–3 shows a boot from a local SCSI drive. The example is abbreviated. For complete instructions on booting *Tru64 UNIX*, see the *Tru64 UNIX Installation Guide*.

Perform the following tasks to boot a system:

1. Power up the system. The system stops at the SRM console prompt, P00>>>.
2. Set boot environment variables, if desired. See Section 6.2.
3. Install the boot medium. For a network boot, see Section 6.3.1.
4. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
5. Enter the **boot** command ❷ and command-line parameters (if you have not set the associated environment variables). In Example 6–3, boot device and boot flags ❸ have already been set.

### 6.3.1 Booting Tru64 UNIX Over the Network

---

To boot the system over the network, make sure the system is registered on a Remote Installation Services (RIS) server. See the *Tru64 UNIX* document entitled *Sharing Software on a Local Area Network* for registration information.

---

#### Example 6-4 RIS Boot

```
P00>>> show device
dka0.0.0.1.1          DKA0          RZ2DD-LS  0306
dka100.1.0.1.1       DKA100       RZ2DD-LS  0306
dka200.2.0.1.1       DKA200       RZ1CB-CS  0844
dkb0.0.0.3.1         DKB0          RZ25      0900
dqa0.0.0.15.0        DQA0          TOSHIBA CD-ROM XM-6302B 1012
dva0.0.0.1000.0      DVA0
eia0.0.0.4.1         EIA0          00-00-F8-09-90-FF
eib0.0.0.2002.1      EIB0          00-06-2B-00-25-5B
pka0.7.0.1.1         PKA0          SCSI Bus ID 7
pkb0.7.0.3.1         PKB0          SCSI Bus ID 7
P00>>> set eia0_protocols bootp
P00>>> set eia0_inet_init bootp
P00>>> boot eia0
.
```

Systems running *Tru64 UNIX* support network adapters, designated ew\*0 or ei\*0. The asterisk stands for the adapter ID (a, b, c, and so on).

1. Power up the system. The system stops at the SRM console prompt, P00>>>.
2. Set boot environment variables, if desired. See Section 6.2.
3. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
4. Enter the following commands. Example 6–4 assumes you are booting from eia0. If you are booting from another drive, enter that device name instead.

```
P00>>> set eia0_protocols bootp
P00>>> set eia0_inet_init bootp
```

The first command ❷ enables the bootp network protocol for booting over the Ethernet controller. The second command ❸ sets the internal Internet database to initialize from the network server through the bootp protocol.

5. Enter the **boot** command ❹ and command-line parameters (if you have not set the associated environment variables). In Example 6–4 the **boot** command sets the system to boot automatically from eia0.

For complete instructions on booting *Tru64 UNIX* over the network, see the *Tru64 UNIX Installation Guide*.

## 6.4 Installing Tru64 UNIX

---

**Tru64 UNIX is installed from the DVD/CD-ROM drive connected to the system.**

---

### Example 6-5 Tru64 UNIX Installation Display

```
P00>>> b dqa0
(boot dqa0.0.0.15.0 -flags a
block 0 of dqa0.0.0.15.0 is a valid boot block
reading 16 blocks from dqa0.0.0.15.0
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 2000
initializing HWRPB at 2000
initializing page table at 1fff0000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

Tru64 UNIX boot - Thu Jan 14 15:03:19 EST 1999

Loading vmunix ...
.
.

Initializing system for Tru64 UNIX installation.  Please
wait...

*** Performing CDRom Installation

Loading installation process and scanning system hardware.

Welcome to the Tru64 UNIX Installation Procedure

This procedure installs Tru64 UNIX onto your system.  You will
be asked a series of system configuration questions.  Until
you answer all questions, your system is not changed in any
way.

During the question and answer session, you can go back to any
previous question and change your answer by entering:  history
You can get more information about a question by entering:
help
```

There are two types of installations:

- o The Default Installation installs a mandatory set of software subsets on a predetermined file system layout.
- o The Custom Installation installs a mandatory set of software subsets plus optional software subsets that you select. You can customize the file system layout.

The Tru64 UNIX Shell option puts your system in single-user mode with superuser privileges. This option is provided for experienced Tru64 UNIX system administrators who want to perform file system or disk maintenance tasks before the installation.

The Installation Guide contains more information about installing Tru64 UNIX.

- 1) Default Installation
- 2) Custom Installation
- 3) Tru64 UNIX Shell

Enter your choice:

- ❶ Boot the operating system from the DVD/CD-ROM drive connected to the system.
- ❷ Follow the *Tru64 UNIX* installation procedure displayed after the installation process is loaded.
  - If your system console is a VGA monitor, the X Server is started and an Installation Setup window is displayed. Click on the fields in the Installation Setup window to enter your responses to the installation procedure.
  - If your system console is a serial terminal, a text-based installation procedure is displayed, as shown in Example 6–5. Enter the choices appropriate for your system.

See the *Tru64 UNIX Installation Guide* for complete installation instructions.

## 6.5 Booting OpenVMS

---

***OpenVMS is booted from a local SCSI disk drive or from a DVD/CD-ROM drive on the InfoServer.***

---

### Example 6-6 Booting OpenVMS from a Local Disk

```
P00>>> show device
dka0.0.0.1.0          DKA0          RZ1DF-BF  1614
dkb0.0.0.3.0          DKB0          COMPAQ BB00911CA0 3B05
dkb100.1.0.3.0        DKB100        COMPAQ BB00911CA0 3B05
dkb200.2.0.3.0        DKB200        COMPAQ BB00911CA0 3B05
dkb300.3.0.3.0        DKB300        COMPAQ BB00911CA0 3B05
dkc0.0.0.4.1          DKC0          COMPAQ BB00911CA0 3B05
.
.
.
P00>>> boot -flags 0,0 dka0
(boot dka0.0.0.1.0 -flags 0,0)
block 0 of dka0.0.0.1.0 is a valid boot block
reading 924 blocks from dka0.0.0.1.0
bootstrap code read in
base = 2c4000, image_start = 0, image_bytes = 73800
initializing HWRPB at 2000
initializing page table at 2bfb6000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

OpenVMS (TM) Alpha Operating System, Version B7.2-1H1



Example 6–6 shows a boot from a local disk. The example is abbreviated. For complete instructions on booting *OpenVMS*, see the *OpenVMS* installation document.

1. Power up the system. The system stops at the SRM console prompt, P00>>>.
2. Set boot environment variables, if desired. See Section 6.2.
3. Install the boot medium. For a network boot, see Section 6.2.4.
4. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
5. Enter the **boot** command and command-line parameters (if you have not set the associated environment variables.). In Example 6–6, the **boot** command with the **-flags** option ❷ causes the system to boot from [SYS0.EXE] on device DKA0.

## 6.5.1 Booting OpenVMS from the InfoServer

---

You can boot *OpenVMS* from a LAN device on the InfoServer. The devices are designated EI\*0 or EW\*0. The asterisk stands for the adapter ID (a, b, c, and so on).

---

### Example 6-7 InfoServer Boot

```
P00>>> show device ❶
dka0.0.0.1.1          DKA0          RZ2CA-LA  N1H0
dka100.1.0.1.1        DKA100        RZ2CA-LA  N1H0
dqa0.0.0.15.0         DQA0          TOSHIBA CD-ROM XM-6302B 1012
dva0.0.0.1000.0       DVA0
eia0.0.0.6.1          EIA0          00-00-F8-10-D6-03
pka0.7.0.1.1          PKA0          SCSI Bus ID 7
P00>>>
.
.
.
P00>>> boot -flags 0,0 -file apb_0712 eia0 ❷
      (boot eia0.0.0.6.1 -file APB_0712 -flags 0,0)
Trying MOP boot.
.....
Network load complete.
Host name: CALSUN
Host address: aa-00-04-00-a4-4e
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 70400
initializing HWRPB at 2000
initializing page table at 3ffee000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

Network Initial System Load Function  
Version 1.2

③

```
FUNCTION          FUNCTION
ID
1      -          Display Menu
2      -          Help
3      -          Choose Service
4      -          Select Options
5      -          Stop
Enter a function ID value:
```

Enter a function ID Value: 3

④

```
OPTION          OPTION
ID
1      -          Find Services
2      -          Enter known Service Name
```

Enter an Option ID value: 2

Enter a Known Service Name: ALPHA\_V71-2\_SSB

OpenVMS (TM) Alpha Operating System, Version V7.1-2

1. Power up the system. The system stops at the P00>>> console prompt.
2. Insert the operating system DVD/CD-ROM into the drive connected to the InfoServer.
3. Enter the **show device** command ❶ to determine the unit number of the drive for your device.
4. Enter the **boot** command and any command-line parameters ❷. In Example 6–7 the device is EIA0. APB\_0712 is the file name of the APB program used for the initial system load (ISL) boot program.

The InfoServer ISL program displays a menu ❸.

5. Respond to the menu prompts ❹, using the selections shown in this example.

For complete instructions on booting *OpenVMS* from the InfoServer, see the *OpenVMS* installation document.

## 6.6 Installing OpenVMS

---

**After you boot the operating system DVD or CD-ROM, an installation menu is displayed on the screen. Choose item 1 (Install or upgrade OpenVMS Alpha). Refer to the *OpenVMS* installation document for information on creating the system disk.**

---

### Example 6-8 OpenVMS Installation Menu

OpenVMS (TM) Alpha Operating System, Version V7.1-2 ❶  
Copyright © 1999 Digital Equipment Corporation. All rights reserved.

Installing required known files...

Configuring devices...

\*\*\*\*\*

You can install or upgrade the OpenVMS Alpha operating system  
or you can install or upgrade layered products that are included  
on the OpenVMS Alpha operating system CD-ROM.

You can also execute DCL commands and procedures to perform  
"standalone" tasks, such as backing up the system disk.

Please choose one of the following:

- 1) Install or upgrade OpenVMS Alpha Version V7.1-2
- 2) Display products that this procedure can install
- 3) Install or upgrade layered products
- 4) Show installed products
- 5) Reconfigure installed products
- 6) Remove installed products
- 7) Execute DCL commands and procedures
- 8) Shut down this system

Enter CHOICE or ? for help: (1/2/3/4/5/6/7/8/?) 1 ❷

- ❶ The *OpenVMS* operating system DVD/CD-ROM is booted.
- ❷ Choose option 1 (Install or upgrade OpenVMS Alpha). To create the system disk, see the *OpenVMS* installation document.



# Chapter 7

## Operation

This chapter gives basic operating instructions. Sections include:

- SRM Console
- Displaying a *Tru64 UNIX* or *OpenVMS* Configuration
- Setting SRM Environment Variables
- Setting SRM Console Security
- Setting Automatic Booting
- Changing the Default Boot Device
- Soft Partitioning
- Hard Partitioning

## 7.1 SRM Console

The SRM console is located in an EEROM on the standard I/O module. From the console interface, you set up and boot the operating system, display the system configuration, and perform other tasks. For complete information on the SRM console, see the *AlphaServer GS80/160/320 Firmware Reference Manual*.

### 7.1.1 SRM Command Overview

Table 7-1 summarizes the most frequently used SRM console commands; Table 7-2 gives the command notation formats; and Table 7-3 shows special characters used on the command line.

**Table 7-1 Summary of SRM Commands**

Command	Function
<b>boot</b>	Loads and starts the operating system.
<b>continue</b>	Resumes program execution on the specified processor or on the primary processor if none is specified.
<b>crash</b>	Forces a crash dump at the operating system level.
<b>edit</b>	Invokes the console line editor on a RAM script or on the user power-up script, "nvram," which is always invoked during the power-up sequence.
<b>help (or man)</b>	Displays information about the specified console command.
<b>init</b>	Resets the SRM console and reinitializes the hardware.
<b>more</b> [filename]	Displays a file one screen at a time.
<b>set <i>envvar</i></b>	Sets or modifies the value of an environment variable.
<b>show <i>envvar</i></b>	Displays the state of the specified environment variable.



**Table 7-1 Summary of SRM Commands (Continued)**

Command	Function
<b>show config</b>	Displays the logical configuration at the last system initialization.
<b>show device</b>	Displays a list of controllers and bootable devices in the system.
<b>show error</b>	Reports errors logged in the EEPROMs.
<b>show fru</b>	Displays the physical configuration of all field-replaceable units (FRUs).
<b>show memory</b>	Displays information about system memory.
<b>show pal</b>	Displays the versions of <i>Tru64 UNIX</i> and <i>OpenVMS</i> PALcode.
<b>show version</b>	Displays the version of the SRM console program installed on the system.
<b>test</b>	Verifies the configuration of the devices in the system.
<b>&lt;esc&gt;&lt;esc&gt;scm</b>	Escape sequence used to enter the SCM console.

**Table 7-2 Notation Formats for SRM Console Commands**

<b>Attribute</b>	<b>Conditions</b>
Length	Up to 255 characters, not including the terminating carriage return or any characters deleted as the command is entered. To enter a command longer than 80 characters, use the backslash character for line continuation.
Case	Upper- or lowercase characters can be used for input. Characters are displayed in the case in which they are entered.
Abbreviation	Only by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously.
Options	You can use command options, to modify the environment, after the command keyword or after any symbol or number in the command. See individual command descriptions for examples.
Numbers	Most numbers in console commands are in decimal notation.
No characters	A command line with no characters is a null command. The console program takes no action and does not issue an error message; it returns the console prompt. The console supports command-line recall and editing.
Spaces or tabs	Multiple adjacent spaces and tabs are compressed and treated as a single space. Leading and trailing spaces are ignored.

**Table 7-3 Special Characters for SRM Console**

Character	Function
Return or Enter	Terminates a command line. No action is taken on a command until it is terminated. If no characters are entered and this key is pressed, the console just redisplay the prompt.
Backslash (\)	Continues a command on the next line. Must be the last character on the line to be continued.
Delete	Deletes the previous character.
Ctrl/A	Toggles between insert and overstrike modes. The default is overstrike.
Ctrl/B or up-arrow	Recalls previous command or commands. The last 16 commands are stored in the recall buffer.
Ctrl/C	Terminates the process that is running. Clears Ctrl/S; resumes output suspended by Ctrl/O. When entered as part of a command line, deletes the current line. Ctrl/C has no effect as part of a binary data stream.
Left-arrow	Moves the cursor left one position.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/F or right-arrow	Moves the cursor right one position.
Ctrl/H	Moves the cursor to the beginning of the line.
Backspace	Deletes ones character.
Ctrl/J	Deletes the previous word.
Ctrl/O	Stops output to the console device for the current command. Toggles between enable and disable. The output can be reenabled by other means as well: when the console prompts for a command, issues an error message, or enters program mode, or when Ctrl/P is entered.

*Continued on next page*

**Table 7-3 Special Characters for SRM Console (Continued)**

Character	Function
Ctrl/Q	Resumes output to the console device that was suspended by Ctrl/S.
Ctrl/R	Redisplays the current line. Deleted characters are omitted. This command is useful for hardcopy terminals.
Ctrl/S	Suspends output to the console device until Ctrl/Q is entered. Cleared by Ctrl/C.
Ctrl/U	Deletes the current line.
*	Wildcarding for commands such as <b>show</b> .
" "	Double quotes enable you to denote a string for environment variable assignment.
#	Specifies that all text between it and the end of the line is a comment. Control characters are not considered part of a comment.

### 7.1.2 Setting the Control Panel Message

**When the operating system is running, the control panel displays the console revision. It is useful to create a customized message if you have a number of systems and you want to identify each system by a node name. Use the set ocp\_text command.**

#### Example 7-1 Set Ocp\_Text Command

```
SCM_E0> set ocp_text nodenamealpha

power on.....
.
.
SCM_EF> sho ocp
Line0: nodenamealpha
Line1: Power ON
Line2: Cpu-10 Mem-6  Pci-6
Line3: Keyswitch-ON
```

## 7.2 Displaying the System Configuration

---

**View the system hardware configuration from the SRM console. It is useful to view the hardware configuration to ensure that the system recognizes all devices, memory configuration, and network connections.**

---

Use the following SRM console commands to view the system configuration.

Additional commands to view the system configuration are described in the *AlphaServer GS80/160/320 Firmware Reference Manual*.

- |                    |  |
|--------------------|--|
| <b>show boot*</b>  | Displays the boot environment variables.   |
| <b>show config</b> | Displays the logical configuration of interconnects and buses on the system and the devices found on them. |
| <b>show device</b> | Displays the bootable devices and controllers in the system.   |
| <b>show fru</b>    | Displays the physical configuration of FRUs (field-replaceable units).                                     |
| <b>show memory</b> | Displays configuration of main memory.   |

### 7.2.1 Show Boot Command

---

**Use the show boot\* command to list the boot environment variables.**

---

#### Example 7-2 Show Boot\*

```
P00>>> show boot*
boot_dev          dka0.0.0.1.1
boot_file
boot_osflags      a
boot_reset        OFF
bootdef_dev       dka0.0.0.1.1
booted_dev
booted_file
booted_osflags
```

## 7.2.2 Show Config Command

Use the show config command to display the entire logical configuration. Example 7-3 shows a GS80 system configuration.

### Example 7-3 Show Config

```
P00>>> sh conf
```

```

                                Compaq Computer Corporation
                                Compaq AlphaServer GS80 6/631

SRM Console      X5.7-1838, built on Dec  1 1999 at 02:02:47
PALcode          OpenVMS PALcode V1.71-2, Tru64 UNIX PALcode V1.64-2

QBB 0            Hard QBB 0
Quad Switch
Duplicate Tag    Up To 4 MB Caches
CPU 0            CPU 0      4 MB Cache
CPU 1            CPU 1      4 MB Cache
CPU 2            CPU 2      4 MB Cache
Memory 0
Memory 1
Memory 2
Memory 3
Directory
IOP
Local Link 0
Remote Link 0
I/O Port 0
PCI Box 0        Riser 0    Right Side
PCI Bus 0        Hose 0     64 Bit, 33 MHz
PCI Bus 1        Hose 1     64 Bit, 33 MHz
Local Link 1
Remote Link 1
I/O Port 1
PCI Box 0        Riser 1    Left Side
PCI Bus 0        Hose 2     64 Bit, 33 MHz
PCI Bus 1        Hose 3     64 Bit, 33 MHz
Local Link 2
Remote Link 2
I/O Port 2
PCI Box 3        Riser 0    Right Side
PCI Bus 0        Hose 4     64 Bit, 33 MHz
PCI Bus 1        Hose 5     64 Bit, 33 MHz
Local Link 3
Remote Link 3
I/O Port 3
PCI Box 3        Riser 1    Left Side
PCI Bus 0        Hose 6     64 Bit, 33 MHz
PCI Bus 1        Hose 7     64 Bit, 33 MHz
Global Port
                                GPA rev 0, GPD revs 0/0

                                QSA rev 2, QSD revs 0/0/0/0
                                DTag revs 1/1/1/1
                                EV67 pass 2.2.2
                                EV67 pass 2.2.2
                                EV67 pass 2.2.2
                                MPA rev 1, MPD revs 1/1
                                MPA rev 1, MPD revs 1/1
                                MPA rev 1, MPD revs 1/1
                                MPA rev 1, MPD revs 1/1
                                DMA rev 1, DMD rev 0
                                IOA rev 2, IOD revs 0/0
                                NE ML rev 2
                                FE ML rev 2
                                PCA rev 3
                                Backplane rev 3
                                PCI rev 2.1 compliant
                                PCI rev 2.1 compliant
                                NE ML rev 2
                                FE ML rev 2
                                PCA rev 3
                                Backplane rev 3
                                PCI rev 2.1 compliant
                                PCI rev 2.1 compliant
                                NE ML rev 2
                                FE ML rev 2
                                PCA rev 3
                                Backplane rev 3
                                PCI rev 2.1 compliant
                                PCI rev 2.1 compliant
                                NE ML rev 2
                                FE ML rev 2
                                PCA rev 3
                                Backplane rev 3
                                PCI rev 2.1 compliant
                                PCI rev 2.1 compliant
                                GPA rev 0, GPD revs 0/0
```

- ❶ **Firmware.** Version numbers of the SRM console, *OpenVMS* PALcode, and *Tru64 UNIX* PALcode.
- ❷ **QBB0.** Components listed include the quad switch and the following modules: CPUs, memory modules, directory module, IOP module, and global port. Chip revision numbers are also listed. Component information for each QBB in the system is displayed.
- ❸ **PCI I/O information, PCI Box 0.** In this example, QBB0 is connected to PCI Box 0 and PCI box 3 (see ❹).

QBB0 I/O port 0 is linked to remote I/O riser 0 located on the right side of PCI box 0. Logical hose numbers are 0 and 1.

QBB0 I/O port 1 is linked to remote I/O riser 1 located on the left side of PCI box 0. Logical hose numbers are 2 and 3.
- ❹ **PCI I/O information, PCI Box 3.** QBB0 is also connected to PCI box 3.

QBB0 I/O port 0 is linked to remote I/O riser 0 located on the right side of PCI box 3. Logical hose numbers are 4 and 5.

QBB0 I/O port 1 is linked to remote I/O riser 1 located on the left side of PCI box 3. Logical hose numbers are 6 and 7.

*Continued on next page*

### Example 7-3 Show Config (Continued)

```

QBB 1                Hard QBB 1
Quad Switch
Duplicate Tag
CPU 1                CPU 5      4 MB Cache
CPU 2                CPU 6      4 MB Cache
Memory 0
Memory 3
Directory
IOP
Local Link 0
Remote Link 0
I/O Port 0
PCI Box 1    Riser 0    Right Side    Backplane rev 3
PCI Bus 0      Hose 8      64 Bit, 33 MHz    PCI rev 2.1 compliant
PCI Bus 1      Hose 9      64 Bit, 33 MHz    PCI rev 2.1 compliant
Local Link 1
Remote Link 1
I/O Port 1
PCI Box 1    Riser 1    Left Side    Backplane rev 3
PCI Bus 0      Hose 10     64 Bit, 33 MHz    PCI rev 2.1 compliant
PCI Bus 1      Hose 11     64 Bit, 33 MHz    PCI rev 2.1 compliant
Local Link 2
Local Link 3
Global Port
NE ML rev 2
FE ML rev 2
PCA rev 3
GPA rev 0, GPD revs 0/0

QBB      Size      Interleave      System Memory 12 GB
0         8 GB      32-Way
1         4 GB      16-Way

Hose  IOP  PCI Box      Primary      Standard I/O Modules
0      0    0
4      0    3
8      1    1
Acer Labs M1543C rev A1-E
Acer Labs M1543C rev A1-E
Acer Labs M1543C rev A1-E

System Memory 12 GB

Board  Set  Array  Size  Address      QBB 0 Memory
0      0    0      1 GB  00000000000
1      0    1      1 GB  00000000000
2      0    2      1 GB  00000000000
3      0    3      1 GB  00000000000
0      1    4      1 GB  00000000000
1      1    5      1 GB  00000000000
2      1    6      1 GB  00000000000
3      1    7      1 GB  00000000000
Total Available      8 GB  32-Way Interleave

Board  Set  Array  Size  Address      QBB 1 Memory
0      0    0      1 GB  01000000000
3      0    3      1 GB  01000000000
0      1    4      1 GB  01000000000
3      1    7      1 GB  01000000000
Total Available      4 GB  16-Way Interleave

```



- ⑤ **QBB1.** QBB1 components are listed.
- ⑥ **PCI I/O information, PCI box 1.** QBB1 is connected to only one PCI box, PCI box 1.  
  
QBB1 I/O port 0 is linked to remote I/O riser 0 located on the right side of PCI box 1. Logical hose numbers are 8 and 9.  
  
QBB1 I/O port 1 is linked to remote I/O riser 1 located on the left side of PCI box 1. Logical hose numbers are 10 and 11.
- ⑦ The total system memory size is reported.  
  
QBB0 has 8 Gbytes in a 32-way interleave; QBB1 has 4 Gbytes in a 16-way interleave.
- ⑧ Standard I/O modules in the system. In this example, PCI box 0, 1, and 3 each has a standard I/O module. The primary standard I/O module is located in PCI box 0. Hose and IOP connections are also listed.
- ⑨ System memory shown by board, set, array, and size. See Section 3.7 for more information.

*Continued on next page*

### Example 7-3 Show Config (Continued)

⑩				①		②	③	④	⑤	⑥
PCI	Box	Riser	Slot	Option		Hose	Bus	Slot	Function	Name
	0	0	1	Standard I/O Module		0				
				+ Acer Labs M1543C		0	0	7		
				+ Acer Labs M1543C IDE		0	0	15		dqa
				+ Acer Labs M1543C IDE		0	0	15		dqb
				+ Acer Labs M1543C USB		0	0	19		
				+ QLogic ISP10x0		0	0	1		pka
0		0	2	DE500-BA Network Con		0	0	2		ewa
0		0	3	ELSA GLoria Synergy		0	0	3		vga
0		0	7	DEGPA-SA		1	0	7		
0		1	1	QLogic ISP10x0		2	0	1		pkb
0		1	2	NCR 53C896		2	0	2	0	pkc
				+ NCR 53C896		2	0	2	1	pkd
0		1	3	09608086		2	0	3	0	
				+ 19608086/0415129A		2	0	3	1	
0		1	6	DEC KZPSA		3	0	6		pke
0		1	7	DEC PCI FDDI		3	0	7		fwa
1		0	1	Standard I/O Module		8				
				+ Acer Labs M1543C		8	0	7		
				+ Acer Labs M1543C IDE		8	0	15		dqe
				+ Acer Labs M1543C IDE		8	0	15		dqf
				+ Acer Labs M1543C USB		8	0	19		
				+ QLogic ISP10x0		8	0	1		pkg
1		0	2	DEC KZPSA		8	0	2		pkh
1		0	3	PowerStorm 350		8	0	3		vgb
1		0	5	DECchip 21154-AA		9	0	5		
				+ DE602-AA		9	2	4		eia
				+ DE602-AA		9	2	5		eib
1		0	7	DEC PCI MC		9	0	7		mca
1		1	1	QLogic ISP10x0		10	0	1		pki
1		1	2	DEC KZPSA		10	0	2		pkj
1		1	3	DEC KZPSA		10	0	3		pkk
1		1	7	DEGPA-SA		11	0	7		
3		0	1	Standard I/O Module		4				
				+ Acer Labs M1543C		4	0	7		
				+ Acer Labs M1543C IDE		4	0	15		dqc
				+ Acer Labs M1543C IDE		4	0	15		dqd
				+ Acer Labs M1543C USB		4	0	19		
				+ QLogic ISP10x0		4	0	1		pkf
3		1	1	DE500-BA Network Con		6	0	1		ewb
3		1	2	PBXNP-AA Token Ring		6	0	2		dwa
3		1	7	001B1011/001B1011		7	0	7		
⑦	⑧			⑨						
Slot	Option				Hose 0, Bus 0, PCI					
1	QLogic ISP10x0				pka0.7.0.1.0			SCSI Bus ID 7	⑩	
					dka0.0.0.1.0			COMPAQ BB00921B91		
					dka400.4.0.1.0			RRD46		
2	DE500-BA Network Con				ewa0.0.0.2.0			08-00-2B-C3-C3-B9		
3	ELSA GLoria Synergy									

### Example 7-3 Show Config (Continued)

```

 7  Acer Labs M1543C                               Bridge to Bus 1, ISA
15  Acer Labs M1543C IDE   dqa.0.0.15.0
                                dqb.0.1.15.0
                                dqa0.0.0.15.0
19  Acer Labs M1543C USB   COMPAQ CDR-8435

    Option                               Hose 0, Bus 1, ISA
    Floppy                             dva0.0.0.1000.0

```

- ⑩ **PCI Box, Riser, Slot.** Each PCI box in the system is identified by a number (0 to F hexadecimal). A system can have a maximum of 16 PCI boxes. The physical location of options in the PCI box are identified by the remote I/O riser (0 or 1), and slot number in the PCI box.
- ① **Option.** Options are listed. Note that the standard I/O module occupies slot 1, on hose 0, riser 0, PCI box 0.
- ② **Hose.** PCI devices connect to logical hoses in the PCI box. A **logical** hose number identifies the logical hose for each device. The firmware assigns the logical hose number (0 to 63). Each PCI box has a maximum of four logical hose numbers.
- ③ **Bus.** The remote I/O riser PCI bus will always be numbered 0. Buses behind device bridges are numbered 1, 2, 3, etc.
- ④ **Slot.** This is the logical slot ID number of each device or controller.
- ⑤ **Function.** On multi-function devices, the function ID is displayed.
- ⑥ **Name.** Device names are listed.
- ⑦ This next section of the **show config** output lists I/O devices by logical order. The logical **Slot** number for each option appears in the first column.
- ⑧ The **Option** name appears next.
- ⑨ **Hose 0, Bus 0, PCI.** In this example, the standard I/O module is logical Slot 1 on PCI Bus 0, Hose 0. See ① for a different presentation of this information.
- ⑩ The devices shown logically are the controllers and devices connected to the controllers. COMPAQ BB00921B91 and RRD46 are SCSI drives attached to controller pka0.

*Continued on next page*

### Example 7-3 Show Config (Continued)

Slot 7	Option DEGPA-SA	Hose 1, Bus 0, PCI	
Slot 1	Option QLogic ISP10x0	Hose 2, Bus 0, PCI	SCSI Bus ID 7
		pkb0.7.0.1.2	RZ1CB-CA
		dkb0.0.0.1.2	RZ1CB-CS
		dkb100.1.0.1.2	RZ1CB-CS
		dkb200.2.0.1.2	COMPAQ BB00911CA0
		dkb400.4.0.1.2	RZ1DF-CB
		dkb500.5.0.1.2	RZ1DF-CF
		dkb600.6.0.1.2	SCSI Bus ID 7
2/0	NCR 53C896	pkc0.7.0.2.2	SCSI Bus ID 7
2/1	NCR 53C896	pkd0.7.0.102.2	Bridge to Bus 2, PCI
3/0	09608086		
3/1	19608086/0415129A		
Slot 6	Option DEC KZPSA	Hose 3, Bus 0, PCI	SCSI Bus ID 7
		pke0.7.0.6.3	RZ1CF-CF
		dke100.1.0.6.3	RZ1CF-CF
		dke200.2.0.6.3	COMPAQ BB00911CA0
		dke300.3.0.6.3	RZ1CF-CF
		dke400.4.0.6.3	00-00-F8-48-56-7A
7	DEC PCI FDDI	fwa0.0.0.7.3	
Slot 1	Option QLogic ISP10x0	Hose 4, Bus 0, PCI	SCSI Bus ID 7
		pkf0.7.0.1.4	RZ1DF-BF
7	Acer Labs M1543C	dkf0.0.0.1.4	Bridge to Bus 1, ISA
15	Acer Labs M1543C IDE	dqc.0.0.15.4	
		dqd.0.1.15.4	
		dqc0.0.0.15.4	TOSHIBA CD-ROM XM-
6302B 19	Acer Labs M1543C USB		
	Option Floppy	Hose 4, Bus 1, ISA	
		dvb0.0.0.1000.4	
Slot 4	Option QSW ELan3	Hose 5, Bus 0, PCI	
Slot 1	Option DE500-BA Network Con	Hose 6, Bus 0, PCI	
2	PBXNP-AA Token Ring	ewb0.0.0.1.6	08-00-2B-C3-C0-38
Slot 7	Option 001B1011/001B1011	Hose 7, Bus 0, PCI	
Slot 1	Option QLogic ISP10x0	Hose 8, Bus 0, PCI	SCSI Bus ID 7
		pkg0.7.0.1.8	RZ1DF-BF
		dkg0.0.0.1.8	SCSI Bus ID 7
2	DEC KZPSA	pkh0.7.0.2.8	RZ1CF-CF
		dkh100.1.0.2.8	RZ1CB-CS
		dkh200.2.0.2.8	COMPAQ BB00911CA0
		dkh300.3.0.2.8	COMPAQ BB00911CA0
		dkh400.4.0.2.8	

①

*Continued on next page*

### Example 7-3 Show Config (Continued)

```

3      PowerStorm 350
7      Acer Labs M1543C
15     Acer Labs M1543C IDE    dge.0.0.15.8      Bridge to Bus 1, ISA
                                dqf.0.1.15.8
                                dge0.0.0.15.8      COMPAQ CDR-8435

19     Acer Labs M1543C USB

        Option                Hose 8, Bus 1, ISA
        Floppy                dvc0.0.0.1000.8

Slot   Option                Hose 9, Bus 0, PCI
5      DECchip 21154-AA      Bridge to Bus 2, PCI
7      DEC PCI MC           Rev: 22, mca0

Slot   Option                Hose 9, Bus 2, PCI
4      DE602-AA             eia0.0.0.2004.9    00-50-8B-6F-2F-92
5      DE602-AA             eib0.0.0.2005.9    00-50-8B-6F-2F-93

Slot   Option                Hose 10, Bus 0, PCI
1      QLogic ISP10x0       pki0.7.0.1.10     SCSI Bus ID 7
2      DEC KZPSA            pkj0.7.0.2.10     SCSI Bus ID 7
                                dkj100.1.0.2.10    RZ1CF-CF
                                dkj200.2.0.2.10    RZ1CB-CS
                                dkj300.3.0.2.10    RZ1CF-CF
                                dkj400.4.0.2.10    RZ1CB-CS
3      DEC KZPSA            pkk0.7.0.3.10     SCSI Bus ID 7

Slot   Option                Hose 11, Bus 0, PCI
7      DEGPA-SA

```

P00>>>

②

- ① Looking back at the information presented under ⑩ helps the user locate the physical PCI slot of each device listed in the logical PCI device output. For example, to find where the DEC KZPSA, Hose 3, Bus 0, PCI device is located, see 3 in the Hose column ②, identify the option by name, and see that it is in PCI box 0, riser 1, slot 6.
- ② The console prompt is displayed.

## 7.2.3 Show Device Command

Use the show device command to display the bootable devices. DK = SCSI drive; DQ = IDE drive; DV = diskette drive; EI or EW = Ethernet controller; PK = SCSI controller.

### Example 7-4 Show Device

```
P00>>> sho dev
dka0.0.0.1.0          DKA0          RZ1DF-BF 1614
dkb0.0.0.7.1          DKB0          COMPAQ BB00911CA0 3B05
dkb100.1.0.7.1        DKB100        COMPAQ BB00911CA0 3B05
dkb200.2.0.7.1        DKB200        COMPAQ BB00911CA0 3B05
dkb300.3.0.7.1        DKB300        COMPAQ BB00911CA0 3B05
dqa0.0.0.15.0         DQA0 ❶ TOSHIBA CD-ROM XM-6302B 1017
dqc0.0.0.15.6         DQC0          TOSHIBA CD-ROM XM-6302B 1017
dqe0.0.0.15.8         DQE0          TOSHIBA CD-ROM XM-6302B 1017
dva0.0.0.1000.0        DVA0
dvb0.0.0.1000.6        DVA0
dvc0.0.0.1000.8        DVA0
dvd0.0.0.1000.16       DVA0
eia0.0.0.3.8          EIA0          08-00-2B-C3-BD-00
fwa0.0.0.4.1          FWA0          00-00-F8-CD-1F-39
fwb0.0.0.5.7          FWB0          00-00-F8-CD-22-E0
fwc0.0.0.1.10         FWC0          00-00-F8-CD-22-FB
pga0.0.0.7.7          PGA0          WWN 1000-0000-c920-da1e
pka0.7.0.1.0          PKA0          SCSI Bus ID 7 5.57
pkb0.7.0.7.1          PKB0          SCSI Bus ID 7 5.57
pkc0.7.0.1.2          PKC0          SCSI Bus ID 7 5.57
```

**Table 7-4 Device Naming Conventions**

<b>①</b>	<b>Category</b>	<b>Description</b>
<b>dq</b>	Driver ID	Two-letter designator of port or class driver dk    SCSI drive or CD       ew    Ethernet port dq    IDE CD-ROM       fw    FDDI device dr    RAID set device       mk    SCSI tape ci    DSSI disk       ci    DSSI tape dv    Diskette drive       pk    SCSI port ei    Ethernet port
<b>a</b>	Storage adapter ID	One-letter designator of storage adapter (a, b, c...).
<b>0</b>	Device unit number	Unique number (MSCP unit number). SCSI unit numbers are forced to 100 X node ID.
<b>0</b>	Bus node number	Bus node ID.
<b>0</b>	Channel number	Used for multi-channel devices.
<b>15</b>	Logical slot number	The slot number assigned by the firmware.
<b>0</b>	Hose number	The hose number assigned by the firmware.

## 7.2.4 Show Memory Command

The show memory command displays the main memory configuration.

### Example 7-5 Show Memory

```
P00>>> show memory
System Memory 128 GB ①
  ②      ③      ④      ⑤      ⑥      ⑦
Board Set Array Size Address QBB 0 Memory
0 0 0 4 GB 00000000000
1 0 1 4 GB 00000000000
2 0 2 4 GB 00000000000
3 0 3 4 GB 00000000000
0 1 4 4 GB 00000000000
1 1 5 4 GB 00000000000
2 1 6 4 GB 00000000000
3 1 7 4 GB 00000000000
Total Available 32 GB 32-Way Interleave ⑧

Board Set Array Size Address QBB 1 Memory
0 0 0 4 GB 01000000000
1 0 1 4 GB 01000000000
2 0 2 4 GB 01000000000
3 0 3 4 GB 01000000000
0 1 4 4 GB 01000000000
1 1 5 4 GB 01000000000
2 1 6 4 GB 01000000000
3 1 7 4 GB 01000000000
Total Available 32 GB 32-Way Interleave

Board Set Array Size Address QBB 2 Memory
0 0 0 4 GB 02000000000
1 0 1 4 GB 02000000000
2 0 2 4 GB 02000000000
3 0 3 4 GB 02000000000
0 1 4 4 GB 02000000000
1 1 5 4 GB 02000000000
2 1 6 4 GB 02000000000
3 1 7 4 GB 02000000000
Total Available 32 GB 32-Way Interleave

Board Set Array Size Address QBB 3 Memory
0 0 0 4 GB 03000000000
1 0 1 4 GB 03000000000
2 0 2 4 GB 03000000000
3 0 3 4 GB 03000000000
0 1 4 4 GB 03000000000
1 1 5 4 GB 03000000000
2 1 6 4 GB 03000000000
3 1 7 4 GB 03000000000
Total Available 32 GB 32-Way Interleave

P00>>>
```



- ❶ The total system memory size is reported.
- ❷ Each memory board (or module) in the QBB is listed. Boards are numbered from 0 to 3. Each QBB can have up to four memory boards.
- ❸ Each memory board has two sets (arrays) of DIMMs installed. A set is numbered 0 or 1. Each set consists of four DIMMs.
- ❹ In this example, all DIMMs have the same density (4 GB) and are in a 32-way interleave. The first array on board 0 is **board 0, set 0**, and is referred to as **array 0**; the second array on board 0 is **board 0, set 1**, and is referred to as **array 4**, and so forth.
- ❺ The size, or density of the array.
- ❻ The starting address of the array.
- ❼ Identifies each QBB in the system.
- ❽ The total available memory in the QBB.

## 7.3 Setting SRM Environment Variables

---

**You may need to set several SRM console environment variables and built-in utilities to configure systems running the *Tru64 UNIX* or *OpenVMS* operating systems.**

---

Set environment variables at the P00>>> prompt.

- To check the setting for a specific environment variable, enter the **show *envar*** command, where the name of the environment variable is substituted for *envar*.
- To reset an environment variable, use the **set *envar*** command, where the name of the environment variable is substituted for *envar*.

For more information on environment variables, see the *AlphaServer GS80/160/320 Firmware Reference Manual*.

## 7.4 Setting SRM Console Security

---

**You can set the SRM console to secure mode to prevent unauthorized personnel from modifying the system parameters or otherwise tampering with the system from the console.**

---

When the SRM is set to secure mode, you can use only two console commands:

- The **boot** command, to boot the operating system
- The **continue** command, to resume running the operating system if you have inadvertently halted the system

The console security commands are as follows:

<b>set password</b>	These commands put the console into secure mode.
<b>set secure</b>	
<b>clear password</b>	Exits secure mode.
<b>login</b>	Turns off console security for the current session.

### 7.4.1 Setting Tru64 UNIX or OpenVMS Systems to Auto Start

---

The SRM **auto\_action** environment variable determines the default action the system takes when the system is power cycled, reset, or experiences a failure.

---

On systems that are factory configured for *Tru64 UNIX* or *OpenVMS*, the factory setting for **auto\_action** is **halt**. The **halt** setting causes the system to stop in the SRM console. You must then boot the operating system manually.

For maximum system availability, **auto\_action** can be set to **boot** or **restart**.

- With the **boot** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed.
- With the **restart** setting, the operating system boots automatically after the SRM **init** command is issued or the Reset button is pressed, and it also reboots after an operating system crash.

To set the default action to **boot**, enter the following SRM commands:

```
P00>>> set auto_action boot
P00>>> init
```

## 7.5 Changing the Default Boot Device

---

You can change the default boot device for *Tru64 UNIX* or *OpenVMS* with the **set bootdef\_dev** command.

---

With the *Tru64 UNIX* or *OpenVMS* operating systems, you can designate a default boot device. You change the default boot device by using the **set bootdef\_dev** SRM console command. For example, to set the boot device to the IDE CD-ROM, enter commands similar to the following:

```
P00>>> show bootdef_dev
bootdef_dev    dka400.4.0.1.1
P00>>> set bootdef_dev dqa500.5.0.1.1
P00>>> show bootdef_dev
bootdef_dev    dqa500.5.0.1.1
```

## 7.6 Soft Partitioning

---

Soft partitioning allows you to run multiple instances of an *OpenVMS* operating system on one hardware system. Soft partitions are created by setting environment variables that define the number of partitions, as well as the CPU modules, I/O risers, memory size, and size of shared memory. Also, one partition is assigned to receive error interrupts. See the *OpenVMS Alpha Galaxy Guide* and *GS80/160/320 Firmware Reference Manual* for more information.

---

### About Soft Partitioning

- Only the *OpenVMS* operating system supports soft partitioning.
- Each partition is maintained by the system management console.
- Each soft partition must have at least one CPU module, one memory module, an I/O riser module, and a PCI box with a standard I/O module.
- Each soft partition requires a standard I/O module with resident SRM firmware in order to boot an operating system copy (or instance).
- Two I/O risers in a QBB cannot be split between partitions.
- CPU and memory resources can be shared across partitions; I/O is bound to a particular partition.

You define what resources are allocated to each partition by setting SRM environment variables. Table 7–5 lists the environment variables used to create soft partitions. After you have assigned the environment variable values, issue the **lpinit** command to initialize the partitions defined.

**Table 7-5 SRM Environment Variables for Soft Partitions**

Environment Variable	Definition
<b>lp_count</b> <i>n</i>	<p>The number of soft partitions to create. Possible values are:</p> <ul style="list-style-type: none"> <li>0 Default. All IOPs, CPUs, and memory are assigned to one soft partition. No shared memory is defined.</li> <li>1 One soft partition is created (partition 0). You need to use the other environment variables to define the partition.</li> <li>2-8 From two to eight soft partitions can be defined. You need to use the other environment variables to define the partitions.</li> </ul>
<b>lp_io_mask</b> * <i>x</i>	For the *, supply the partition number (0 – 7). The value <i>x</i> gives a binary mask indicating which QBBs (containing I/O risers) are included in the partition.
<b>lp_cpu_mask</b> * <i>x</i>	For the *, supply the partition number. The value <i>x</i> gives a binary mask indicating which CPUs are part of the partition.
<b>lp_mem_size</b> * <i>n</i>	For the *, you supply the number of the partition. The value <i>n</i> can be a size or list of sizes on a 64 MB boundary.
<b>lp_shared_mem_size</b> <i>n</i>	Size of memory shared by all partitions or a list of sizes on an 8 MB boundary.
<b>lp_error_target</b>	The number of the soft partition that is to receive error interrupts. The primary CPU of this soft partition will receive the interrupts. If <b>lp_count</b> is 0, <b>lp_error_target</b> is ignored.

At the SRM console prompt, you set values for one environment variable to define the number of soft partitions in the system, one to set the memory mode, and two for each partition that define the CPU and I/O modules in each partition. The **lpinit** command completes the procedure.

### Example 7-6 Defining Soft Partitions

P00>>> set lp_count 3	❶
P00>>> set lp_io_mask0 1	❷
P00>>> set lp_cpu_mask0 f	❸
P00>>> set lp_mem_size0 6GB	
P00>>> set lp_io_mask1 6	❹
P00>>> set lp_cpu_mask1 cf0	
P00>>> set lp_mem_size1 12GB	
P00>>> set lp_io_mask2 8	❺
P00>>> set lp_cpu_mask2 f300	
P00>>> set lp_mem_size2 10GB	
P00>>> set lp_shared_mem_size 4GB	❻
P00>>> set lp_error_target 0	❼
P00>>> lpinit	❽

- ❶ The number of soft partitions is set to 3.
- ❷ The **set lp\_io\_mask0 1** command defines QBB0 (and its I/O risers) as residing in partition 0.
- ❸ **Set lp\_cpu\_mask0 f** assigns CPUs 0–3 to partition 0. The **set lp\_mem\_size0 6GB** command assigns 6 GB of memory to partition 0.
- ❹ **Set lp\_io\_mask1 6** defines QBB1 and QBB2 (and its I/O risers) as residing in partition 1. The **set lp\_cpu\_mask1 cf0** command assigns CPUs 4–7, 10, 11 to partition 1. The **set lp\_mem\_size1 16GB** command assigns 12 GB of memory to partition 1.
- ❺ The **set lp\_io\_mask2 8** defines QBB3 (and its I/O risers) as residing in partition 2. **Set lp\_cpu\_mask1 f300** assigns CPUs 8, 9, 12–15 to partition 2. **Set lp\_mem\_size1 10GB** assigns 10 GB of memory to partition 2.
- ❻ The **set lp\_shared\_mem\_size 4GB** command assigns 4 GB of memory to be shared by all partitions.
- ❼ **Set lp\_error\_target 0** assigns partition 0 to receive and process the error interrupts.
- ❽ The **lpinit** command completes the procedure and partitions the system.

As a final result of the commands entered in Example 7–6, you will have the following soft partition setup:

Instance	CPU	Memory	I/O
0	0, 1, 2, and 3	6 GB	QBB 0
1	4, 5, 6, 7, 10, and 11	12 GB	QBB 1 and QBB 2
2	8, 9, 12, 13, 14, and 15	10 GB	QBB 3
		4 GB (shared)	

## 7.7 Hard Partitioning

Hard partitioning allows you to run multiple operating systems on one hardware system. Table 7-6 lists the SCM environment variables used to create hard partitions. After entering the SCM commands, set the control panel switch to Off and then to On to create the hard partitions.

Table 7-6 SCM Environment Variables for Hard Partitions

Environment Variable	Definition
<b>hp_count</b>	A value from 0 to $n$ indicating the number of hard partitions to be configured. Partitions are numbered from 0 to $n-1$ . If this value equals 0 or 1, partitioning is disabled and the system remains in standard SMP mode.
<b>hp_qbb_mask<math>x</math></b>	A bitmask of QBBs to be included in hard partition $x$ .

In Example 7-7, a GS160 system is hard partitioned into 2 partitions with:

- QBB 0 and QBB 1 in partition 0, and,
- QBB 2 and QBB 3 in partition 1

### Example 7-7 Defining Hard Partitions

```
SCM_E0> set hp_count 2
SCM_E0> set hp_qbb_mask0 3
SCM_E0> set hp_qbb_mask1 c
```



Partitioning a system is done at the QBB level: You can have a maximum of eight partitions. In hard partitioning mode:

- System partitions are independent.
- Each partition requires its own configuration tree.
- Hardware isolation is required.
- Address spaces are disjointed.
- Failures are isolated to a specific partition.



# Chapter 8

## Using the System Control Manager

The system control manager (SCM) communicates with microprocessors on the console serial bus (CSB) to monitor and manage the system. The SCM also provides remote server management functions.

This chapter explains the operation and use of the SCM. Sections are:

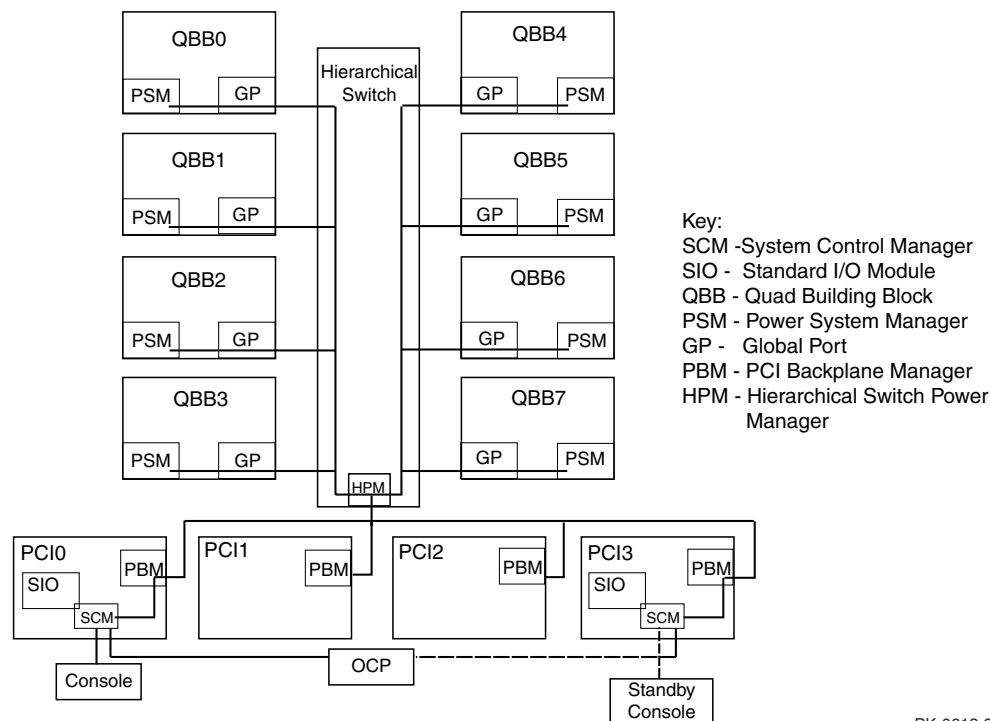
- Console Serial Bus Subsystem
- System Control Manager
- SCM COM1 Operating Mode
- Console Device Setup
- Entering the SCM
- SRM Environment Variables for COM1
- SCM Command-Line Interface
- Troubleshooting Tips

For more information on SCM commands, see the *AlphaServer GS80/160/320 Firmware Reference Manual*.

## 8.1 Console Serial Bus Subsystem

The console serial bus (CSB) links microprocessors throughout the system, forming a monitoring and control subsystem managed by the system control manager (SCM). The SCM microprocessor is located on the standard I/O module in the master PCI box. Figure 8-1 shows a block diagram of the CSB, the microprocessors (or “managers”), and the SCM. Also shown is a redundant SCM available as a standby console.

Figure 8-1 CSB Block Diagram



PK-0613-98

The SCM communicates with the PCI backplane managers (PBMs), the hierarchical switch power manager (HPM), and the QBB power system managers (PSMs) distributed throughout the CSB subsystem. The subsystem has a power source separate from the rest of the system called auxiliary voltage (Vaux). Vaux enables the subsystem to be turned on from a remote site when the main system breaker is on and the system is plugged in.

## 8.2 System Control Manager Overview

---

**With the SCM, you can monitor and control the system (reset, power on/off, halt) without using the operating system. You can enter SCM commands at a local or remote console device. You also manage hard partitions using the SCM.**

---

The SCM:

- Controls the control panel display and writes status messages on the display.
- Detects conditions that require alerts, such as excessive temperature, fan failure, and power supply failure. On detection, SCM displays messages on the OCP, pages an operator, and sends an interrupt to the SRM, which then passes the interrupt to the operating system or an application.
- Retrieves and passes information about a system shutdown to the SRM console at the next power-up. SRM displays a message regarding the last shutdown.
- Provides a command-line interface (CLI) for the user to control the system. From the CLI the user can power the system on and off, halt or reset the system, and monitor the system environment.
- Passes error log information to shared RAM so that this information can be accessed by the system.
- Handles power supply hot swap.
- Manages hard partitions using HP\_\* environment variables.
- Shared RAM on every standard I/O module in the system is kept coherent by the master SCM. This allows every SCM console and copy of the operating system in a hard-partitioned system to see the same configuration information.

## **SCM Firmware**

SCM firmware resides on the standard I/O module in flash memory. If the SCM firmware should ever become corrupted or obsolete, you can update it manually using the Loadable Firmware Update Utility. The microprocessor can also communicate with the system power control logic to turn on or turn off power to the rest of the system.

The SCM is powered by an auxiliary 5V supply. You can gain access to the SCM as long as AC power is available to the system (through the wall outlet) and the control panel keyswitch is in the On position. Thus, if the system fails, you can still access the SCM and gather information about the failure.

## **PSMs, PBMs, and HPMs**

The power system managers (PSMs), PCI backplane managers (PBMs), and hierarchical switch power manager (HPM) are the “slaves” of the master SCM. The SCM gathers all environmental data from the slaves. The PSMs, PBMs, and HPM perform the following:

- Monitors environmental sensors
- Monitors voltages, power supplies, and fans
- Shuts down a subsystem (QBB or PCI box) if a fatal condition (temperature reaches a fatal limit or a power failure occurs) is detected

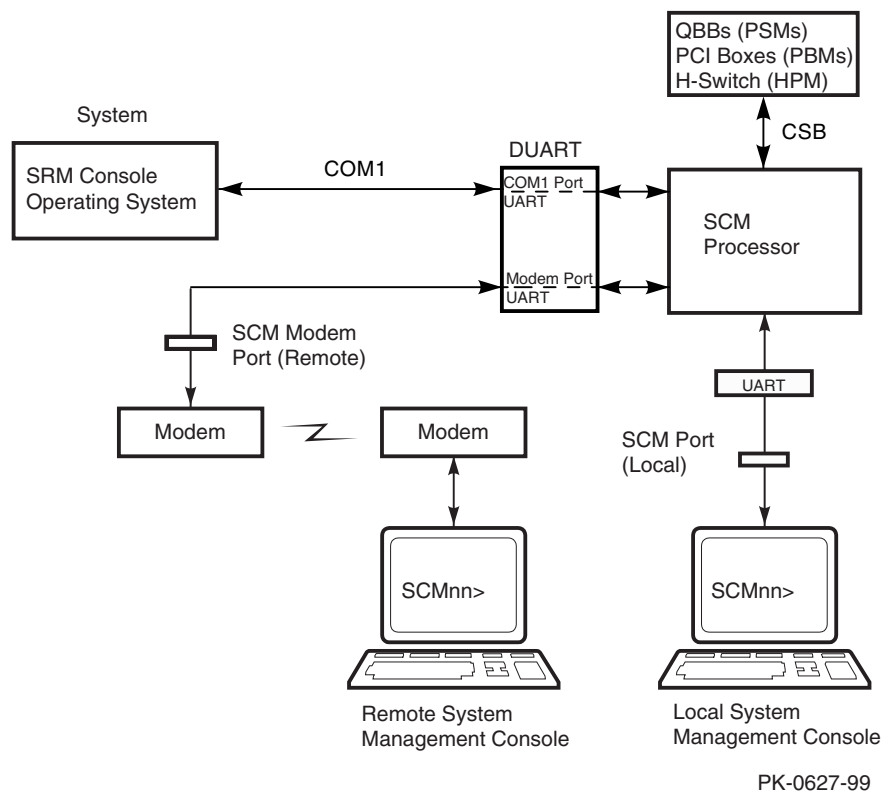
## **Configuration, Error Log, and Asset Information**

The SCM provides additional functionality to read and write configuration and error log information to FRU error log devices. All reading and writing of the FRU EEPROMs is done by the slaves under the direction of the SCM.

## 8.3 SCM COM1 Operating Modes

The SCM can be configured to manage different data flow paths defined by the `com1_mode` environment variable. In through mode (the default), all data and control signals flow from the system COM1 port through the SCM to the active external port. You can also set bypass modes so that the signals partially or completely bypass the SCM. The `com1_mode` environment variable can be set from either SRM or the SCM. See Section 8.7.1.

Figure 8-2 Data Flow in Through Mode





## Through Mode

Through mode is the default operating mode. The SCM routes every character of data between the internal system COM1 port and the active external port, either the local port or the modem port. If a modem is connected, the data goes to the modem. The SCM filters the data for a specific escape sequence. If it detects the default escape sequence, “<Esc><Esc>scm”, it enters the SCM CLI.

Figure 8–2 illustrates the data flow in through mode. The internal system COM1 port is connected to one port of the DUART chip, and the other port is connected to an external modem port, providing full modem controls. The DUART is controlled by the SCM microprocessor, which moves characters between the two UART ports. The local port is always connected to the internal UART of the microprocessor. The escape sequence signals the SCM to enter the CLI. Data issued from the CLI is transmitted between the SCM microprocessor and the active port that enters the SCM.

---

**NOTE:** *The internal system COM1 port should not be confused with the external COM1 serial port on the standard I/O module. The internal COM1 port is used by the system software to send data either to the COM1 port on the system or to the SCM modem port if a modem is connected.*

---

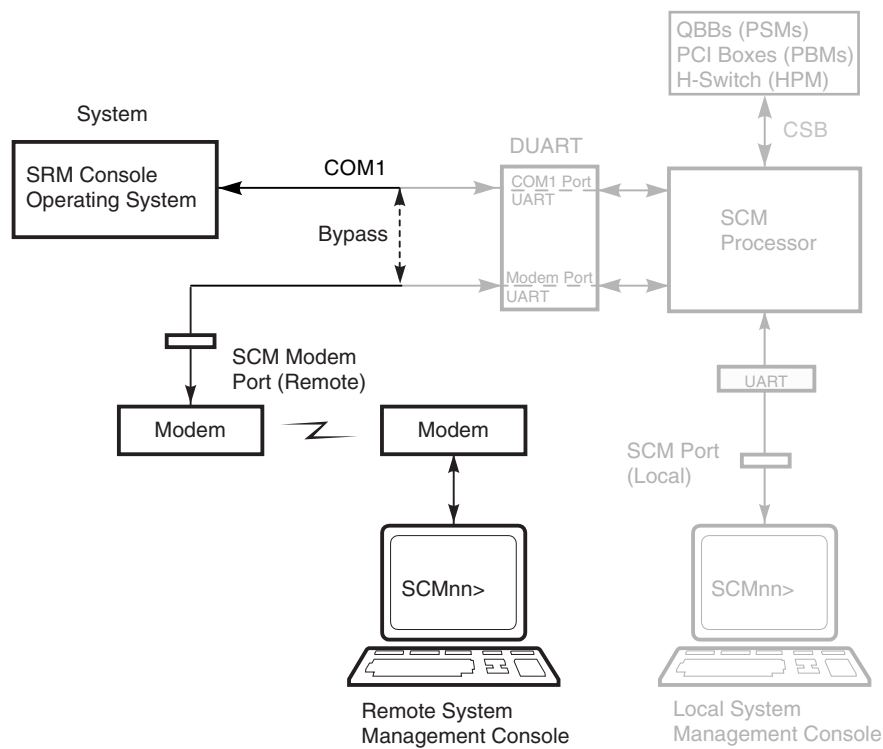
## Local Mode

You can set a local mode in which only the local channel can communicate with the system COM1 port. In local mode the modem is prevented from sending characters to the system COM1 port, but you can still enter the SCM from the modem.

### 8.3.1 Bypass Modes

For modem connection, you can set the operating mode so that data and control signals partially or completely bypass the SCM. The bypass modes are snoop, soft bypass, and firm bypass.

Figure 8–3 Data Flow in Bypass Mode



PK-0627A-99

Figure 8–3 shows the data flow in bypass mode. Note that the internal system COM1 port is connected directly to the modem port.

---

**NOTE:** *You can connect a console device to the modem port in any of the bypass modes.*

---

The local console device is still connected to the SCM and can still enter the SCM to switch the COM1 mode if necessary. In any of the bypass modes, when the system loses power, it defaults to snoop mode.

### **Snoop Mode**

In snoop mode data partially bypasses the SCM. The data and control signals are routed directly between the system COM1 port and the external modem port, but the SCM taps into the data lines and listens passively for the SCM escape sequence. If it detects the escape sequence, it enters the SCM CLI.

The escape sequence is also passed to the system on the bypassed data lines. If you decide to change the default escape sequence, be sure to choose a unique sequence so that the system software does not interpret characters intended for the SCM.

In snoop mode the SCM is responsible for configuring the modem for dial-in as well as dial-out alerts and for monitoring the modem connectivity.

Because data passes directly between the two UART ports, snoop mode is useful when you want to monitor the system but also ensure optimum COM1 performance.

### **Soft Bypass Mode**

In soft bypass mode all data and control signals are routed directly between the system COM1 port and the external modem port, and the SCM does not listen to the traffic on the COM1 data lines. The SCM is responsible for configuring the modem and monitoring the modem connectivity. If the SCM detects loss of carrier, it switches automatically into snoop mode. If you have set up the dial-out alert feature, the SCM pages the operator if an alert is detected and the modem line is not in use.

Soft bypass mode is useful if management applications need the COM1 channel to perform a binary download, because it ensures that SCM does not accidentally interpret some binary data as the escape sequence.

*Continued on next page*

After downloading binary files, you can set the **com1\_mode** environment variable from the SRM console to switch back to snoop mode or other modes for accessing the SCM, or you can hang up the current modem session and reconnect it.

### **Firm Bypass Mode**

Firm bypass mode effectively disables the SCM. All data and control signals are routed directly from the system COM1 port to the external modem port. The SCM does not configure or monitor the modem, and the SCM dial-in and call-out features are disabled. Firm bypass mode is useful if you want the system, not the SCM, to fully control the modem port and you want to disable SCM remote management features. You can switch to other modes, if necessary, by resetting the **com1\_mode** environment variable from the SRM console.

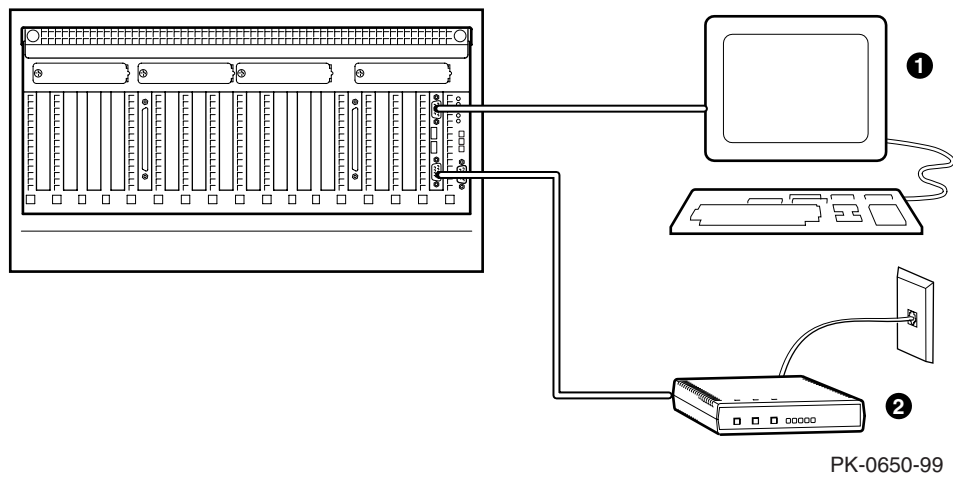
### **Hard Bypass Mode**

Hard bypass mode is set by a jumper on the standard I/O module. If this jumper is set, then all other modes are prohibited and the SCM is only accessible from the local port.

## 8.4 Console Device Setup

You can use the SCM from a console device connected to the system or a modem hookup. As shown in Figure 8-4, local connection is made from the local port ❶. You connect remotely from a modem connected to the modem port ❷.

Figure 8-4 Setups for SCM (PCI Box)



Note: Both the local port and the modem port are set at:  
1 start bit, 8 data bits, 1 stop bit, no parity.

Local port defaults: baud rate is 9600, flow control is soft.

Modem port defaults: baud rate is 57600, flow control is hard.

## 8.5 Entering the SCM

---

Type an escape sequence to invoke the SCM from the SRM console. You can enter SCM from a modem, or from a local console device. From a VGA monitor, you can enter SCM commands from the SRM console.

---

- You can enter the SCM from the local console device regardless of the current operating mode.
- You can enter the SCM from the modem if the SCM is in through mode, snoop mode, or local mode. In snoop mode the escape sequence is passed to the system and displayed.

### Entering from a Console Device

Invoke the SCM from a console device by typing the following default escape sequence (six plus signs):

```
<Esc><Esc>scm
```

To exit, enter the **quit** command. This action returns you to whatever you were doing before you invoked the SCM. In the following example, the **quit** command returns you to the system COM1 port.

```
SCM_E0> quit  
Returning to COM1 port
```

### Entering SCM Commands from a VGA Console

From the SRM console, enter:

```
P00>>> SCM <CLI_command>
```

## 8.6 SRM Environment Variables for COM1

---

Several SRM environment variables allow you to set up the COM1 port for use with the SCM. Default values are read from shared RAM and set to whatever the SCM values are at console boot time.

---

<b>com1_baud</b>	Sets the baud rate of the COM1 port and the modem port. The default is <b>57600</b> for the local port and the modem port. Other allowable baud rates are: 38400, 19200, 9600, 7200, 4800, 3600, and 2400.
<b>com1_flow</b>	Specifies the flow control on the serial port. The default is <b>software</b> . Other values include <b>hardware</b> , <b>none</b> , and <b>both</b> .
<b>com1_mode</b>	Specifies the COM1 data flow paths so that data either flows through the SCM or bypasses it. This environment variable can be set from either the SRM or the SCM. The default value is <b>through</b> . Other values are <b>local</b> , <b>snoop</b> , <b>soft bypass</b> , and <b>firm bypass</b> .
<b>com1_modem</b>	Specifies to the operating system whether or not a modem is present.

## 8.7 SCM Command-Line Interface

The system control manager supports setup commands and commands for managing the system. See Table 8-1. For an SCM commands reference, see the *AlphaServer GS80/160/320 Firmware Reference Manual*.

Table 8-1 SCM Commands

Command	Description
<b>Clear {alert, error}</b>	Clears firmware or hardware state (alert). Clears SDD and TDD error logs (error).
<b>Crash</b>	Forces a crash dump at the operating system level.
<b>Disable {alert, remote}</b>	Disable system alerts or remote access.
<b>Enable {alert, remote}</b>	Enables system alerts or remote access.
<b>Erase</b>	Erases the current screen.
<b>Fault</b>	Emulates the control panel Fault button.
<b>Halt</b>	Halts the operating system.
<b>Hangup</b>	Terminates a remote session, disconnecting the modem.
<b>Help</b>	Provides SCM command information
<b>Init</b>	Prepares modem to send alerts and receive incoming calls.
<b>Power</b>	Directs the system control manager to supply power to the processors, memories, and I/O modules in the system.
<b>Quit</b>	Exits the SCM command-line interpreter and returns control to the COM1 port.
<b>Reset</b>	Resets the system.
<b>Set alert</b>	Specifies a string that defines the time to wait for the remote pager to respond to a dial-out, and a message to be sent to the remote pager.



**Table 8-1 SCM Commands (Continued)**

<b>Command</b>	<b>Description</b>
<b>Set modem_baud</b>	Sets the modem baud rate.
<b>Set dial</b>	Sets the string to be used by the SCM to dial out when an alert condition occurs.
<b>Set local_baud</b>	Sets the baud rate of the SCM-to-local console device UART.
<b>Set com1_baud</b>	Sets the baud rate to the SCM-to-system UART.
<b>Set hp_count</b>	Defines the number of hard partitions in system.
<b>Set hp_qbb_mask<math>x</math></b>	Sets a value that creates a binary mask in which a bit set to 1 indicates that a QBB belongs to a partition $x$ .
<b>Set init</b>	Assigns the modem initialization string appropriate for the dial-out modem at the system site.
<b>Set password</b>	Defines the password to be used by the remote user when dialing into the system.
<b>Show csb</b>	Displays a list of all known nodes on the console serial bus.
<b>Show fru</b>	Displays information on field-replaceable units (FRUs).
<b>Show nvr</b>	Displays additional nonvolatile environment variables.
<b>Show status</b>	Displays SCM status and information.
<b>Show system</b>	Displays an overall system summary, including the most recent system self-test results and the current temperature and voltage readings.
<b>Test alert</b>	Tests the modem setup and connection to the remote pager or computer.

## Command Conventions

Follow these conventions when entering SCM commands:

- Enter enough characters to distinguish the command.

---

**NOTE:** *The **reset** and **quit** commands are exceptions. You must enter the entire string for these commands to work.*

---

- For commands consisting of two words, enter the entire first word and at least one letter of the second word. For example, you can enter **disable a** for **disable alert**.
- For commands that have parameters, you are prompted for the parameter.
- Use the Backspace key to erase input.

## Power-Up Messages

- A \*\*\* message is a diagnostic error format displayed when an error is detected by SROM or XSROM code. Power-up continues depending on the type of error and system configuration.
- A ### message is a diagnostic warning format detected by SROM or XSROM code. System power-up continues, and the resource remains available for use.
- An ~E~ message indicates a more severe error has been detected; power-up continues, but the affected resource is dropped.
- A ~W~ message indicates an error has been reported to the SCM by a “slave” microprocessor (PSM, PBM, or HPM). Power-up continues, but the affected resource is questionable.
- A ~I~ message is informational only, and indicates no error condition.

### 8.7.1 Defining the COM1 Data Flow

---

Use the set **com1\_mode** command from SRM or SCM to define the COM1 data flow paths.

---

You can set **com1\_mode** to one of the following values:

<b>through</b>	All data passes through SCM and is filtered for the escape sequence. This is the default.
<b>snoop</b>	Data partially bypasses SCM, but SCM taps into the data lines and listens passively for the escape sequence.
<b>soft_bypass</b>	Data bypasses SCM, but SCM switches automatically into snoop mode if loss of carrier occurs.
<b>firm_bypass</b>	Data bypasses SCM. SCM remote management features are effectively disabled.
<b>local</b>	Changes the focus of the COM1 traffic to the local 9-pin port if SCM is currently in one of the bypass modes or is in through mode with an active remote session.

---

#### Example

```
SCM_E0> set com1_mode snoop
```

## 8.7.2 Displaying the System Status

The **SCM status** command displays the current SCM settings. Table 8-2 explains the status fields.

```
SCM_E0> show status

      System Management Settings

SCM escape sequence      : ^]^[scm
Local Baud/flow control  : 57600 / soft
COM1 Baud/flow control   : 57600 / hard
Modem Baud flow control  : 57600 / hard
COM1 mode                : Pass-through
OCP power switch        : Off
OCP halt                 : Deasserted
OCP secure               : Non-Secure
Remote access            : Disabled
Remote user              : Not Connected
Alerts                   : Disabled
Modem password           : wffirmware
Modem init string        : ate0v0&c1s0=2
Modem dial string        : atdt915085551212
Modem alert string       : , , , , , , , 5085551212#;
Alert pending            : No
Most recent alert        :
SCM_E0>
```

**Table 8-2 Status Command Fields**

Field	Meaning
SCM escape sequence:	Currently defined escape sequence that returns you to the SCM console. Default is "<Esc><Esc>scm".
Local Baud/flow control	Baud rate of the local port.
COM1 Baud/flow control	Baud rate of the system COM1 port.
Modem Baud/flow control	Baud rate of the system modem.

**Table 8-2 Status Command Fields (Continued)**

<b>Field</b>	<b>Meaning</b>
OCP power switch: OCP halt: OCP secure:	Indicates the position of the control panel keyswitch and Halt button. In this case, the keyswitch is in the Off position and the Halt button is not pushed in. The OCP is in a nonsecure state.
Remote access:	Remote access to the system is disabled or enabled.
Remote user:	A remote user is either not connected, as in the example, or connected.
Alerts:	Alert status is reported: Enabled or Disabled. Possible alerts are listed in Section 8.7.6.
Modem password:	The modem password is displayed.
Modem init string:	Modem initialization string set by the <b>set init</b> command.
Modem dial string:	Modem dial string, set by the <b>set dial</b> command.
Modem alert string:	Modem alert string, set by the <b>set alert</b> command.
Alert pending:	Yes = Alert has been triggered. No = No alert has been triggered.
Most recent alert:	Tells the cause of the most recent alert.

### 8.7.3 Displaying the System Environment

The **SCM show system** command provides a snapshot of the system environment.

```
SCM_E0> show system
```

①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
Par	hrd/csb QBB#	CPU	Mem	IOR3 IOR2 IOR1 IOR0 (pci_box.rio)	GP	QBB	Dir	PS	Temp
		3210	3210		Mod	BP	Mod	321	(°C)
(-)	0/30	PPPP	PPPP	Px.x P2.0 Pf.1 Pf.0	P	P	P	PPP	33.5
(-)	1/31	PPPP	PPPP	P4.0 Px.x Px.x Px.x	P	P	P	PPP	31.0
(-)	2/32	PPPP	PPPP	Px.x P0.0 Px.x P3.0	P	P	P	PP-	29.0
(-)	3/33	PPPP	PPPP	--.- --.- Px.x P5.0	P	P	P	PP-	33.5

①	②	③	④
HSwitch	Type	Cables	Temp (°C)
HPM40	4-port	7 6 5 4 3 2 1 0	28.0

⑤	⑥	⑦	⑧
PCI Rise1-1 Rise1-0 Rise0-1 Rise0-0 Cab 7 6 5 4 3 2 1 7 6 5 4 3 2 1	RIO	PS	Temp
	1 0	21	(°C)
10 - - - - - L - L - - S	- *	-P	31.5
12 - - - - - - - - L S	- *	P-	31.5
13 - - - - - - L - - S	- *	P-	31.5
14 - - - - - - L - L S	- *	PP	31.5
15 - - - - L - - - - S	- *	PP	31.0
1f - - - - - - - H L - S	* *	PP	31.5

```
SCM_E0>
```

- ① In a partitioned system, the number shown in this column identifies the hard partition that each QBB is a member of.
- ② The hard QBB and soft QBB numbers are listed. Location in the system determines hard QBB numbers; the system firmware assigns soft QBB numbers.
- ③ CPU self-tests are reported. Note that there are four CPUs in each of the four QBBs. In this example, all 16 CPUs have passed self-test as indicated by a P.
- ④ Memory self-tests are reported. The 16 memory modules have passed self-test.
- ⑤ Remote I/O riser map for each QBB.

For QBB0:

Px.x indicates that a remote I/O riser is present (P), but no I/O mapping (x.x) has been determined.

P2.0 indicates that a hose from local I/O riser 2, port 2, (IOR2) is connected to PCI box 2, remote I/O riser 0.

Pf.1 indicates that a hose from local I/O riser 1, port 1 (IOR1) is connected to PCI box F, remote I/O riser 1.

Pf.0 indicates that a hose from local I/O riser 1, port 0 (IOR0) is connected to PCI box F, remote I/O riser 0.

- ⑥ Global port module self-test results.
- ⑦ QBB backplane test results.
- ⑧ Directory module self-test results.
- ⑨ Power supply test results.
- ⑩ Power supply temperature readings.
- ① HSwitch. The model number of the H-switch is displayed. In this example the model number is HPM40.
- ② Type. The HPM40 is a 4-port H-switch.
- ③ Cables. In the example, H-switch ports 0 through 3 are enabled (P); H-switch ports 4 through 7 are not present (–). Other indicators are V (valid), \* (present), and F (failed).
- ④ Temp. Displays the temperature inside the H-switch housing.
- ⑤ PCI Cab. This map shows the presence or absence of devices in each PCI box in the system. Each PCI box has a maximum of four PCI hoses (two hoses (0 and 1) on remote I/O riser 0 and two hoses (0 and 1) on remote I/O riser 1.

In this example, PCI box 10 has a standard I/O module (S) in riser 0, hose 0, slot 1, and two low-powered devices (L) in riser 0, hose 1, slots 4 and 6.

A device is labeled according to its power usage: L is low-powered, M is medium-powered, and H is high-powered.

*Continued on next page*

- ⑥ RIO. Remote I/O modules. \* indicates the presence of a remote I/O module; – indicates its absence. In the example, PCI box 10 has one remote I/O riser, I/O riser 0, installed.
- ⑦ PS. PCI box power supplies 2 and 1. A **P** indicates a power supply is powered on and passed self-test; a – indicates no power supply is installed. Other indications include:
  - p**, power supply passed self-test and is powered off
  - \***, power supply is present, but no pass/fail status yet
  - F**, power supply failed
- ⑧ Temp. Displays the temperature on the PCI backplane.

#### 8.7.4 Power On and Off, Reset, and Halt

---

**The SCM **power on**, **power off**, **halt in**, **halt out**, and **reset** commands perform the same functions as the buttons on the control panel.**

---

##### Power On and Power Off

The SCM **power on** command powers the system on, and the **power off** command powers the system off. The Power button on the control panel, however, has precedence.

- If the system has been powered off with the Power button, the SCM cannot power the system on. If you enter the **power on** command, the message “Power button is OFF” is displayed, indicating that the command will have no effect.
- If the system has been powered on with the Power button, and the **power off** command is used to turn the system off, you can toggle the Power button to power the system back on.



## Halt In and Continue

The **halt in** command halts the operating system. The **continue** command releases the halt.

Issuing the **continue** command will restart the operating system even if the Halt button is latched in.

## Reset

---

**NOTE:** *The environment variable, `auto_quit`, must be enabled for the **reset** command to return to the console or operating system.*

---

The SCM **reset** command restarts the system. The console device exits SCM and reconnects to the server's COM1 port.

```
SCM_E0> reset
Returning to COM port
```

## 8.7.5 Configuring Remote Dial-In

---

**Before you can dial in through the SCM modem port or enable the system to call out in response to system alerts, you must configure the system for remote dial-in.**

---

### Example 8-1 Dial-In Configuration

```
SCM_E0> enable remote          ❶
SCM_E0> set password wffirmware ❷
SCM_E0> set init ate0v0&c1s0=2 ❸
SCM_E0> init                   ❹

Querying the modem port...modem detected
Initializing modem...passed
Modem port initialized
SCM_E0>                        ❺
```

---

**NOTE:** *The following modems require the initialization strings shown here. For other modems, see your modem documentation.*

---

Modem	Initialization String
Compaq Microcom 510 (North America)	Ate0v0&c1s0=2
3Com U. S. Robotics Courier V.Everything/56K (North America and Japan)	Ate0v0&c1s0=2
Hayes Accura (North America)	Ate0v0&c1&k3s0=2
AIWA PV-BW5610 56K V.90 & K56flex Data/Fax Modem (Japan)	Ate0v0&c1s0=2
OMRON ME5614D (Japan)	Ate0v0&c1s0=2
MultiTech MT5600ZDX (North America)	Ate0v0&c1s0=2
MultiTech MT5600ZDXe (Europe)	

- ❶ Enables remote access to the SCM modem port.
- ❷ Sets the password (in the example, “wffirmare”) that is prompted for at the beginning of a modem session. The string cannot exceed 14 characters and is not case sensitive. For security, the password is not echoed on the screen. When prompted for verification, type the password again.
- ❸ Sets the modem initialization string. The string is limited to 31 characters and can be modified depending on the type of modem used.

The SCM automatically configures the modem's flow control according to the setting of the SRM **com1\_flow** environment variable. The SCM also enables the modem carrier detect feature to monitor the modem connectivity.

After the initialization string is set, it is copied to the SCM's EEPROM to allow the SCM to reinitialize the modem in the case of a system power cycle.

- ❹ The **init** command is entered to initialize the modem.
- ❺ At this point, the modem is configured for remote dial-in.

### Dialing In

Once the remote session is connected, the password must be entered to access the SCM CLI. The following example shows the screen output when a modem connection is established.

```
atdt915085551212
CONNECT 9600
```

```
Password: *****
Access granted
```

```
Modem escape sequence verified
SCM_E0>
```

1. At the SCM\_E0> prompt, enter SCM CLI commands to monitor and control the system. Use the **show status** command to check the current management settings and the channel the modem connection is currently using.
2. When you have finished a modem session, enter the **hangup** command to cleanly terminate the session and disconnect from the server.

## 8.7.6 Configuring Alert Dial-Out

Set a few parameters to configure alert dial-out. The dial string and alert string are set to send a message to a remote operator. When an alert condition is triggered, the dial string is sent first, followed by the alert string. Once the two strings are set, alert dial-out is enabled and the new parameters should be verified. The elements of the dial string and alert string are shown in Table 8-3.

You must configure remote dial-in for the dial-out feature to be enabled.

To set up the dial-out alert feature, enter the SCM from the local console device.

### Example 8-2 Alert Dial-Out Configuration

```
SCM_E0> set dial atdt915551212 ❶

SCM_E0> set alert ,,,,5085551234#; ❷

SCM_E0> enable alert ❸
Enabling alerts
SCM_E0> test alert ❹
SCM_E0> show status ❺

System Management Settings

RMC escape sequence      : <Esc><Esc>scm
Local BAUD/flow control  : 57600 / soft
COM1 BAUD/flow control   : 57600 / hard
Modem BAUD/flow control  : 57600 / hard
COM1 Mode                : Pass-through
OCP power switch         : Off
OCP HALT                 : Deasserted
OCP secure               : Non-Secure
Remote access            : Enabled
Remote user              : Not Connected
Alerts                   : Enabled
Modem password           : wffirmware
Modem init string        : ate0v0&c1s0=2
Modem dial string        : atdt915551212
Modem alert string       : ,,,,5085551234#;
Alert pending            : YES ❻
Most recent alert        : User initiated ... from CSB node e0

SCM_E0> clear alert ❼
Pending alert cleared
SCM_E0>
```

- ❶ Sets the string to be used by the SCM to dial out when an alert condition occurs. The dial string must include the appropriate modem commands to dial the number.
- ❷ Sets the alert string that is transmitted through the modem when an alert condition is detected. Set the alert string to the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem.
- ❸ Alert dial-out is enabled.
- ❹ Issue the **test alert** command to test the alert dial-out settings.
- ❺ Now issue the **show status** command to verify settings. Note that the alert pending value is YES (❻), indicating the test alert was successful.
- ❼ Once the dial-out settings are verified, the pending alert should be cleared by entering the **clear alert** command.

Paging services vary, so you need to become familiar with the options provided by the paging service you will be using. The SCM supports only numeric messages.

---

**NOTE:** *If you do not want dial-out paging enabled at this time, enter the **disable alert** command after you have tested the alert dial-out function. Alerts continue to be logged, but no paging occurs.*

---

#### Alert Conditions Reported to a Remote Operator

Alert Condition Detected By:	Alert Condition
PSM (power system manager)	Blower failure Power failure Overtemperature condition
HPM (H-switch power manager)	Overtemperature condition Power failure System clock failure
PBM (PCI backplane manager)	Fan failure Overtemperature condition Power supply failure
User	User-initiated alert

**Table 8-3 Elements of Dial String and Alert String**

<b>Dial String</b>	
	The dial string is case sensitive. The SCM automatically converts all alphabetic characters to uppercase.
ATDT	AT = Attention. D = Dial T = Tone (for touch-tone)
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it. ,= Pause for two seconds.
15551212	Phone number of the paging service.
<b>Alert String</b>	
,,,,,	Each comma (,) provides a 2-second delay. In this example, a delay of 8 seconds is set to allow the paging service to answer.
5085551234#	A call-back number for the paging service. The alert string must be terminated by the pound (#) character.
;	A semicolon must be used to terminate the entire string.

### 8.7.7 Resetting the Escape Sequence

---

The **SCM set escape** command allows the user to change the escape sequence. The default escape sequence is “<Esc><Esc>scm”.

---

The new escape sequence can be any printable character string, not to exceed six characters. Use the **show status** command to verify the new escape sequence.

```
SCM_E0> set escape
Escape Sequence: foofoo
SCM_E0> show status
.
.
.
RMC Escape Sequence: foofoo
```

## 8.8 Troubleshooting Tips

Table 8-4 lists possible causes and suggested solutions for symptoms you might see.

Table 8-4 SCM Troubleshooting

Symptom	Possible Cause	Suggested Solution
You cannot enter the SCM from the modem.	The SCM may be in soft bypass or firm bypass mode.	Issue the <b>show com1_mode</b> command from SRM and change the setting if necessary. If in soft bypass mode, you can disconnect the modem session and reconnect it.
The console device cannot communicate with the SCM correctly.	System and console device baud rates do not match.	Set the baud rate for the console device to be the same as for the system. For first-time setup, suspect the console device, since the SCM and system default baud is 9600.
SCM will not answer when the modem is called.	Modem cables may be incorrectly installed.	Check modem phone lines and connections.
	SCM remote access is disabled or the modem was power cycled since last being initialized.	From the local console device, enter the <b>set password</b> and <b>set init</b> commands, and then enter the <b>enable remote</b> command.
	The modem is not configured correctly.	Modify the modem initialization string according to your modem documentation.



**Table 8-4 SCM Troubleshooting (Continued)**

Symptom	Possible Cause	Suggested Solution
SCM will not answer when modem is called.	On power-up, SCM defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initializations.	Wait 30 seconds after powering up the system and SCM before attempting to dial in.
After the system is powered up, the COM1 port seems to hang and then starts working after a few seconds.	This delay is normal.	Wait a few seconds for the COM1 port to start working.
New escape sequence is forgotten.		At the SRM prompt, enter the following: P00>> SCM show status  The escape sequence is shown in the output.  Or, reset the system to factory defaults.



# Appendix A

## Jumpering Information

This appendix contains jumpering information for the PCI backplane, the hierarchical switch power manager (HPM), and the standard I/O module.

### A.1 PCI Backplane Jumpers

**Table A-1 lists PCI backplane jumpers and their functions. These two jumpers are not normally installed.**

**Table A-1 PCI Backplane Jumpers**

Jumper	Function
J60	If flash ROM is corrupt, installing this jumper will force the PCI backplane manager (PBM) into fail-safe loader mode. The jumper station is three-pronged and the installed position is indicated by a silkscreen connection.
J62	For future use.

## A.2 HPM Jumpers

**Table A-2 lists hierarchical switch power manager (HPM) jumpers and their functions. The HPM module has three 2-position jumpers, none of which are normally installed.**

**Table A-2 HPM Jumpers**

Jumper	Function, When Installed
J2	Flash_Write_Inhibit. Prevents the hardware from writing to flash memory.
J3	Force_FSL. Causes the firmware to remain in the fail-safe loader code after HPM reset.
J4	HS_CSB_ID0. Sets ID0 of the microprocessor's CSB address field to a 1.

## A.3 Standard I/O Module Jumpers

**Table A-3 lists standard I/O module jumpers and their functions. The jumper station is three-pronged and the installed position is indicated by a silkscreen connection.**

**Table A-3 Standard I/O Module Jumpers**

Jumper	Function, When Installed
J12	Forces SRM console defaults.
J13	Forces COM1 bypass.
J17	SCM remains in fail-safe loader code upon reset.
J18	Forces the SCM defaults.

# Glossary

<b>AC off state</b>	One of the system power states in which all power is removed from the system. See also Hot-swap, Cold-swap, and Warm-swap states.
<b>Clock splitter module</b>	Module that provides the system with multiple copies of the global and I/O reference clocks.
<b>Cold-swap state</b>	One of the system power states in which AC power and Vaux are present in the system, but power is removed from the area being serviced. See also AC off, Hot-swap, and Warm-swap states.
<b>Console serial bus</b>	See CSB.
<b>CSB</b>	Console serial bus; a monitoring and control subsystem, managed by the system control manager (SCM), and which provides a command language used by the operator.
<b>Directory module</b>	Module on the QBB that monitors memory requests and holds the location of the accurate copy of memory data requested. A memory directory module is not needed in a system with only one QBB. In a system with more than one QBB, each QBB needs a directory module.
<b>Distribution board</b>	The interface between the global ports of two QBBs.
<b>Expander cabinet</b>	A cabinet that holds additional storage shelves and PCI boxes.
<b>Global port</b>	Module that plugs into a QBB that provides the interface between the QBB and the hierarchical switch.

<b>Hard partition</b>	A partition whose basic unit is a QBB and which shares no resources with any other QBB; defined by using the SCM command language. The smallest hard partition is one QBB; the maximum number of hard partitions in one system is eight.
<b>Hierarchical switch</b>	See H-switch.
<b>Hose</b>	The connection between a QBB and a PCI box; can be logical or physical.
<b>Hot-swap state</b>	One of the power states of the system in which all power is present in the system. In this state certain components of the system can be replaced. See also Cold-swap, Warm-swap, and AC off states.
<b>HPM</b>	H-switch power manager; controls power and monitors the temperature inside the H-switch housing. The HPM reports status to the system control manager.
<b>H-switch</b>	Hierarchical switch; a high-speed switch connection for all QBBs. Each QBB connects to the hierarchical switch through its global port.
<b>H-switch power manager</b>	See HPM.
<b>I/O port</b>	Backplane logic that provides an interface to the I/O riser module.
<b>I/O riser module</b>	See Local I/O riser module and Remote I/O riser module.
<b>Instance</b>	In a partitioned system, a copy of an operating system.
<b>LFU</b>	Loadable Firmware Update Utility; firmware used to update the SRM console firmware and I/O device firmware.
<b>Loadable Firmware Update Utility</b>	See LFU.
<b>Local and remote</b>	Terminology relating to testing, CPUs, and system management.

<b>Local I/O riser module</b>	The I/O riser module that is on the QBB backplane.
<b>Local primary CPU</b>	The CPU chosen to be the primary CPU in a QBB.
<b>Local testing</b>	Testing confined to the QBB on which the CPU doing the testing resides.
<b>Memory directory module</b>	See Directory module.
<b>OCP</b>	Operator control panel; used by the operator to control the system. It has a keyswitch, display screen, indicators, and buttons. The keyswitch is used to power the system up or down or to secure it from remote access. The screen displays messages during booting. Indicators show system state. The buttons reset or halt the system.
<b>Partition</b>	A subset of a system's CPUs, memory, and I/O that runs its own copy (instance) of an operating system. See also Hard partition, Soft partition.
<b>PBM</b>	PCI backplane manager; monitors and manages the environment and components in the PCI box in which it resides.
<b>PCI backplane manager</b>	See PBM.
<b>PCI expansion box</b>	A PCI box without a standard I/O module, console ports, and storage devices.
<b>PCI master box</b>	A PCI box containing a standard I/O module plus console ports and storage devices.
<b>PCI box</b>	An enclosure that provides 14 PCI slots (four PCI buses, two four-slot buses and two three-slot buses). A master PCI box contains a standard I/O module with the system control manager microprocessor, console ports, storage devices, and logic for the control panel. For redundancy, a secondary PCI master box also has these capabilities. PCI expansion boxes do not have the standard I/O module, console ports, storage devices, and control panel logic.

<b>Power cabinet</b>	Cabinet in the GS160/320 systems that provides power for the system cabinets and houses PCI boxes and storage shelves.
<b>Power input</b>	Power is supplied to the entire system box through the QBB at the rear of the cabinet. Vaux and 48 volt input and return are supplied.
<b>Power system manager</b>	See PSM.
<b>Primary PCI box</b>	The PCI master box whose system control manager (SCM) controls the system. See also Secondary PCI box.
<b>PSM</b>	Power system manager; a module in each QBB that monitors CPUs, voltages, temperatures, and fan speeds in the QBB and then reports this information to the system control manager (SCM). The SCM can make requests of the power system manager.
<b>QBB</b>	Quad building block; the basic building block of the system. The QBB backplane provides a crossbar switch and connectors for one to four processor modules, one to four memory modules, power converters, a power system manager, a memory directory module, a clock splitter module, a global port, and I/O ports.
<b>QBB, hard</b>	Number that indicates the physical location of a QBB in a system.
<b>QBB, soft</b>	Number that indicates a QBB as reported by the firmware.
<b>Rack cabinet</b>	The GS80 cabinet configured as a complete system that holds up to two system drawers, each of which has one QBB.
<b>Remote I/O riser module</b>	The I/O riser module that is in the PCI box.
<b>Remote testing</b>	Testing run by the system primary CPU on a QBB other than the QBB in which it resides.



<b>SCM</b>	System control manager; a microprocessor on a standard I/O module that monitors and controls other microprocessors that monitor system state. The SCM provides a command language for an operator and allows for remote management of the system. A second standard I/O module with another SCM provides a backup control system.
<b>Secondary PCI box</b>	A PCI master box that serves as a backup to the primary PCI box. See also Primary PCI box.
<b>SIO</b>	Standard I/O module. Module in the master PCI box that has I/O ports and the system control manager (SCM) microprocessor.
<b>Soft partition</b>	A partition whose CPU and memory resources can be allocated from any QBB; defined by using the SRM console. Also referred to as logical partitions.
<b>Standard I/O module</b>	See SIO.
<b>System box</b>	The GS160 and 320 enclosure for two quad building blocks (QBBs).
<b>System cabinet</b>	The GS160 and GS320 cabinet that holds up to two system boxes, each of which has two QBBs.
<b>System control manager</b>	See SCM.
<b>System drawer</b>	The GS80 enclosure for one quad building block (QBB).
<b>System management console</b>	The PC used to manage the system.
<b>System primary CPU</b>	The CPU chosen out of all CPUs in all QBBs to be the primary CPU of the system.
<b>Vaux</b>	Low voltage power present in the system even when 48 volt power is not. Vaux powers the console serial bus, allowing the system control manager to monitor the system.

**Warm-swap state**

One of the power states of the system in which power is removed from a specified QBB but other segments of the system remain fully powered. In this state only the power going to the specified QBB is removed so that the QBB can be serviced. See also Hot-swap, Cold-swap, and AC off states.

# Index

## A

- APB program, 6-25
- Architecture, 1-5
- auto\_action environment variable, 7-21
- Auxiliary power supply, SCM, 8-5

## B

- Boot flags
  - OpenVMS, 6-13
  - Tru64 UNIX, 6-12
- Boot procedure
  - OpenVMS, 6-23
  - Tru64 UNIX, 6-17
- boot\_file environment variable, 6-11
- boot\_osflags environment variable, 6-12
- bootdef\_dev environment variable, 6-10
- bootp protocol, 6-15
- Bypass modes, 8-8

## C

- Clock splitter module, 2-15
- COM1 data flow, defining, 8-17
- Command conventions, SCM, 8-16
- Console device setup (SCM), 8-11
- Console serial bus, 8-2
- Control panel switch, 2-28
- CPU module, 2-10

## D

- Device naming, 7-17
- Dial-in configuration, 8-24
- Dial-out alert, 8-26
- Directory module, 2-12
- Drawer
  - characteristics, 4-2
- DUART ports, 8-7

## E

- ei\*\_protocols environment variable, 6-15
- ei\*0\_inet\_init environment variable, 6-14
- env command (SCM), 8-20
- Environment, monitoring, 8-20
- Escape sequence (SCM), 8-12
- ew\*0\_inet\_init environment variable, 6-14
- ew\*0\_protocols environment variable, 6-15

## F

- Fault pushbutton, 2-29
- Firm bypass mode, 8-10
- Firmware updates, 1-4

## G

- Global port module, 2-17

## H

- Halt pushbutton, 2-29
- Halt, remote, 8-23
- hangup command (SCM), 8-25
- Hard partitioning, 7-26
- Hierarchical switch power manager, 8-3

## I

- I/O riser module, 2-16
- InfoServer, 6-24
- ISL boot program, 6-25

## **L**

Loadable firmware update utility, 1-4

## **M**

Memory module, 2-11

Message conventions, SCM, 8-16

MOP protocol, 6-15

## **O**

OpenVMS

booting from InfoServer, 6-24

## **P**

Pagers, 8-27

Partitioning

hard, 7-26

soft, 7-22

PCI backplane manager, 8-3

PCI box

configuration guidelines, 3-20

PIC processor, 8-5

Power modules, 2-13

Power system manager, 8-3

Power system manager module, 2-14

Power-on/off, from SCM, 8-22

## **Q**

QBB

backplane, 2-8

color code, 3-12

power system manager, 8-3

quit command (SCM), 8-12

## **R**

Remote power-on/off, 8-22

Reset pushbutton, 2-29

Reset, from SCM, 8-23

RIS boot procedure, 6-19

## **S**

SCM

auxiliary power supply, 8-5

bypass modes, 8-8

command conventions, 8-16

configuring call-out, 8-24

console device setup, 8-11

data flow diagram, 8-6

dial-out alert, 8-26

env command, 8-20

escape sequence, 8-12

exiting, 8-12

firm bypass mode, 8-10

hangup command, 8-25

logic, 8-5

message conventions, 8-16

overview, 8-4

PIC processor, 8-5

quit command, 8-12

remote power on/off, 8-22

remote reset, 8-23

set com1\_mode command, 8-17

snoop mode, 8-9

status command, 8-18

through mode, 8-7

troubleshooting, 8-30

SCM mode, entering, 8-12

set com1\_mode command (SCM), 8-17

set escape command (SCM), 8-29

Setting SRM environment variables, 7-19

show boot command, 7-7

show command (SRM), 7-7

show config command, 7-8

show device command, 7-16

show memory command, 7-18

Snoop mode, 8-9

soft partitioning, 7-22

SRM console, 1-4

command syntax, 7-4

device naming conventions, 7-17

soft partitioning, 7-22

special characters, 7-5

SRM console commands

show boot, 7-7

show config, 7-8

show device, 7-16

show memory, 7-18

status command (SCM), 8-18

- System box
  - characteristics, 2-2
  - configuration rules, 3-11
  - QBB, 2-6
- System drawer
  - electrical and environmental parameters, 4-3

## **T**

- Through mode (SCM), 8-7
- Troubleshooting
  - SCM, 8-30

## **U**

- Updating firmware, 1-4